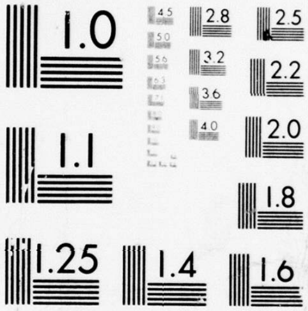
AD-A068 160 SYRACUSE UNIV NY SCHOOL OF COMPUTER AND INFORMATION --ETC F/G 12/1
A DYNAMIC PRIORITY SCHEDULING RULE WITH OPTIMIZATION FOR TRANSA--ETC(U)
FEB 79 D K WOOD F30602-77-C-0235

UNCLASSIFIED RADC -TR-79-8 NL

1 OF 2
AD A068160



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

12

B.S.

RADC-TR-79-8
Interim Report
February 1979

A DYNAMIC PRIORITY SCHEDULING RULE WITH OPTIMIZATION FOR TRANSACTION PROCESSING SYSTEMS

Syracuse University

Daniel K. Wood

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-79-8 has been reviewed and is approved for publication.

APPROVED: Clement Falzarano

CLEMENT FALZARANO
Project Engineer

APPROVED: Wendall C Bauman

WENDALL C. BAUMAN, Colonel, USAF
Information Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 18

RADC-TR-79-8

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER 9

4. TITLE (and Subtitle) 6

A DYNAMIC PRIORITY SCHEDULING RULE WITH OPTIMIZATION FOR TRANSACTION PROCESSING SYSTEMS.

5. TYPE OF REPORT & PERIOD COVERED

Interim Report,
Oct 77 - Oct 78

6. PERFORMING ORG. REPORT NUMBER

N/A

7. AUTHOR(s) 10

Daniel K. Wood

8. CONTRACT OR GRANT NUMBER(s) 15

F30602-77-C-0235 New

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Syracuse University
School of Computer & Information Science
Syracuse NY 13210

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

62702F
16 55811903 17 19

11. CONTROLLING OFFICE NAME AND ADDRESS

Rome Air Development Center (ISIS)
Griffiss AFB NY 13441

12. REPORT DATE 11

Feb[redacted]979

13. NUMBER OF PAGES

156

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

Same 12 165p.

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Clement Falzarano (ISIS)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer	Queues
Transaction Processing	Performance Modeling
Scheduling Algorithms	Optimization
Operating System	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the investigation of a dynamic priority scheduling rule and its applicability for scheduling transaction processing systems. In this report once the scheduling rule was chosen, various cost models were optimized using this rule. Simulation studies were also conducted using this rule. The results from this analysis indicates that this scheduling rule can be applied favorably to transaction processing systems.

UNCLASSIFIED 410176

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

mt

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(*When Data Entered*)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(*When Data Entered*)

ABSTRACT

In this dissertation, we investigate the use of a dynamic priority scheduling rule proposed by Kleinrock in a M/G/1 queueing model and study its applicability as a scheduling rule for transaction processing systems.

In this rule, each job has a priority index and it increases linearly, starting from zero when the job arrives, at a rate which is assigned to the job's priority class. When the server becomes free, it selects the job for service whose priority index is the largest. The rates for the priority classes are a set of parameters that can be varied to control the waiting times of jobs in each priority class.

To study this rule, we first derived the feasible performance space of mean waiting times for each class of jobs. Then we derived an algorithm for determining the values of the control parameters for any given set of feasible mean waiting times. We then discuss optimization of several cost models using functions of mean waiting times. For each cost function, a procedure to obtain an optimal combination of feasible mean waiting times is presented. This set of mean waiting times can then be used to obtain optimal values for the control parameters.

At saturated congestion, we proved a multiplicative version of Jackson's conjecture which states that the ratios among different classes of mean waiting times are proportional to the reciprocals of the control parameters.

Simulation studies of the variance behavior show that as discrimination among different classes increases, the variance of waiting times for each class of jobs varies and it either strictly increases or strictly decreases following the same direction as the mean waiting time changes.

A sensitivity study of mean waiting times shows that the relative waiting times among different classes of jobs remains fairly steady under reasonable fluctuations of system congestion.

The results obtained from our study show that this scheduling rule is applicable for use in transaction processing systems. The capability of adjusting the control parameters to provide mean waiting times performance between that of first-come-first-serve and fixed priority scheduling allows the designer to select from a wide range of response times for different classes of transactions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

CONTENTS (Continued)

CHAPTER 1

TRANSACTIONS PROCESSING SYSTEMS

1.0 Introduction

Over the past few years, a very pronounced trend of increased usage of on-line computer systems has developed in data processing. One type of on-line system that is proliferating in use is Transaction Processing System. Transaction processing systems generally handle an organization's operational data as the data are generated and needed by on-going operations. This method of processing data reduces clerical data handling, gives an almost instant turnaround time and provides current and accurate information whenever and wherever needed.

Simply stated, transaction processing systems are a special type of computer information systems designed for non-expert users to communicate with the computers for on-line processing of transactions. The computer handles transaction workloads by running a set of previously stored application programs to interact with a centrally managed data base. Such systems are found in a vast variety of organizations, including airline, banking, medical, insurance, manufacturing and government.

One representative application of transaction processing systems is the airline reservation system [KNIG72]. Initially designed solely for controlling seat inventory and maintaining limited passenger records [PERR61], airline systems have now evolved into large, complex systems that maintain waiting lists, provide flight information, handle provisions for special facilities such as hotel reservation and car rental, perform load and trim calculations prior to take-off, and even interact with each other for inter-airline flight information.

Since transaction processing applications usually involve a large volume of transactional data for processing, scheduling rules that provide fast response times and efficient usage of computer systems resources are desirable.

This dissertation discusses a time dependent priority scheduling rule for transaction processing systems (TPS) which we call escalating priority scheduling. In this scheduling rule, a set of control parameters can be adjusted by the designer to provide discriminations between different priority classes of transactions. This rule was first proposed by Kleinrock [KLEI64].

In investigating this scheduling rule, we study the behavior and limitations of the rule, derive the feasible performance space of mean waiting times allowed, and discuss some cost models for optimizing this scheduling rule.

In optimizing, our solution procedure is divided into the following two stages: First, for a given objective function, we determine the optimal mean waiting times for each priority class. Second, given a desirable performance of mean waiting times that is feasible, we determine the values of the control parameters that will realize this desirable performance.

In the remainder of this chapter, Section 1.1 gives an introduction to transaction processing systems, Section 1.2 discusses the modeling of transaction processing systems as queueing systems, Section 1.3 reviews and summarizes priority scheduling rules that are relevant to our research, and Section 1.4 defines the problems specific to escalating priority scheduling, and gives a more detailed overview of the dissertation work.

1.1 Transaction Processing Systems

There are certain common characteristics in all transaction processing (TP) environments. They are the following [BOOT72]:

1. The system is highly user oriented. Data entry is through terminals which are located convenient to the users. The computer for processing transactions may be installed in a different location.

2. The terminal operating procedure for data entry is built around "non-expert" users, and is usually quite simple.

3. The input data items, called transactions, are of pre-defined type. Transaction volume is usually large, and transaction applications may be diversified.

4. Application programs to process the transactions are prepared and stored in the computer in advance and are invisible to the user.

5. The response time is fast. The system is capable of providing the user with the desired information within seconds.

6. The system generally maintains an integrated data base. Processing of each transaction usually involves some manipulation -- adding, updating, retrieving or deleting -- of data in the data base.

For transaction processing applications, sophisticated software is designed to meet the following needs:

1. Simple and effective interactive procedures for an efficient flow of information between terminal operators and the computer.

2. Effective utilization of system resources to ensure satisfactory performance.

3. Easy and fast implementation of application programs for system service enhancements.

4. Protective method of data handling to ensure data base integrity and security.

5. Efficient and effective error recovery procedure to maintain system availability.

The current state-of-the-art software systems designed to meet this need are known by several different names: transaction (processing) monitor, teleprocessing monitor, and transaction processing (operating) executive. Some of the better known transaction processing monitors that are available for general use are: CICS from IBM [IBM73], TIP from Univac, TPE from Honeywell [HONE73], TASK/MASTER from Turnkey Systems, SHADOW II from Cullinane Corporation, INTERCOMM from Informatics Inc., and ENVIRON/1 from Cincom Systems, Inc.

These software systems are designed with a high degree of generality so that users can tailor the system toward their specific applications. Generally, these systems can perform the following tasks [IBM73]:

. Terminal management - Host and control data transfer between a telecommunication network of heterogeneous mix of terminals and application programs.

. Multitask control - Schedule and support concurrent processing of a number of application programs for the wide mixture of transaction workloads.

. File management - Provide efficient access methods in handling data base files. Support scheduling and initiation of all file item requests made by the application programs.

. Storage management - Allocate and control of storage for programs, storage for input/output buffering **areas** and temporary storage of data.

. Program management - Provide a multiprogramming capability while offering a real-time program fetch capability. Intercept program interrupts to prevent total system termination.

. Error handling - Detect and handle error conditions caused by hardware or software malfunction. Provide dump facility to assist in analysis of programs and transaction undergoing development or modification.

In the literature, there **is little** detailed documentation of transaction processing systems. Topics of some existing articles discussing transaction processing systems are: a design methodology for TPS [HIRC75], TP facilities in IBM IMS/VS systems [MCGE77], development of a high performance TPS and its monitor [SIWI77], designing customized systems using TP monitors [DAVE74], running TP monitor in a special operating system environment [EADE77], and report of a successful use of TP monitors [GERA76]. In this last article which describes a magazine/ book/ record system of Time, Inc. using CICS of IBM, it is reported that the system is capable of supporting 200 CRT terminals to process 750,000 transactions per week which interact with a data base of five billion characters, while providing a mean response time of one second.

For the evaluation and selection of commercially available transaction processing monitors, Matheny et al. [MATH77] has compiled a list of evaluation criteria. They are: (a) terminal support, (b) internal facilities consideration, (c) application programming considerations, (d) communication functions, (e) operating environment, (f) implementation / maintenance, (g) recovery / reliability / controls, and (h) costs. They have also provided a comparative study of the internal characteristics of the following five transaction processing monitors: CICS, INTERCOMM, TASK/MASTER, ENVIRON/1 and SHADOW II.

1.2 Modeling of Transaction Processing Systems

To date, there are a limited number of documented studies on the design and performance evaluation of transaction processing systems. [GERK74], [SCHW77] and [SARZ77] are three of the articles in the literature that specifically address these problems. Computer performance evaluations are commonly used for the design, selection and tuning (peaking) of computer systems. Modeling of computer systems as queueing systems is a common and desired approach for the evaluation of systems performance. This approach of studying computer timesharing systems has proved to be successful in improving the design, control and effectiveness of systems operation [KLEI76].

Transaction processing systems are very similar to general timesharing systems. Both kinds of systems are designed for on-line use and man-machine interactions require similar constraints on the systems responsiveness. However, certain characteristics that distinguish these two types of systems have limited the usefulness of models that were developed for timesharing systems in the study of transaction processing systems.

Some of the characteristics unique to transaction processing systems are the following:

(a) For each transaction generated from the terminal, the required application program(s) for processing this transaction are stored in the system in advance. Hence the general requirements of system resources needed to process this transaction are known.

(b) Based on the intrinsic characteristics of different types of transactions, their response time requirements may vary. That is, some types of transactions are more important, and need to be processed with priority. On the other hand, other types of transactions may be delayed slightly without significantly affecting the terminal operations.

(c) Due to the nature of transactions, processing logic of the application programs are generally quite straightforward, and the processing path of programs generally do not change with different values of transaction data.

(d) The processing times required for all types of transactions are usually no more than one second. Thus, in a transaction processing environment, processing overhead becomes important. The technique of swapping programs in and out of memory, which is widely used in timesharing systems, is generally not applicable in transaction processing.

In studying queueing systems, the following essential elements need to be characterized:

1. Source population - population from which the entities demanding service emanate.

2. Arrival process - the pattern by which the entities arrive to the service facility.

3. Queue structure - Number and configuration of waiting lines.

4. Service facilities - Number of service channels and ordering of services.

5. Service process - The required time to completely service an entity at each service station.

6. Scheduling discipline - The rule by which units are selected for service.

We now discuss each of these characteristics in the transaction processing environments:

(1) Source population. In many TPS, there are a large number of users. In this case, infinite populations can be used to describe the source population of arrival. However, some TPS may only have a small number of terminals. Then, a finite arrival population is needed for modeling the queueing arrival.

(2) Arrival process. In general, arriving transactions from different terminals are independently generated. Studies of timesharing systems have shown that when a large number of terminals are active (signed on), then Poisson arrivals can be assumed [ANDE74]. Simulations of stochastic processes have shown that when there are 10 or more active terminals, the superimposed arrival process is sufficiently close to a Poisson process even though individual arrivals from each terminal may not be Poisson [AGRA75]. However, when the arrival population is small, some other process may

be needed to describe the arrival process.

(3) Queue structure. Depending upon the type of computer, the number of priority classes, and the scheduling rule, different queue structures are required. Generally, for priority scheduling, it is desirable to maintain separate queues for each priority class for the use of each system resource. Since each transaction, once generated, will need to be processed eventually, an infinite waiting line should be allowed in TPS.

(4) Service facilities. In uniprogramming environments, CPU and I/O operations are alternatively active. Thus the two services together can be thought of as a single service. In this case, TPS can be modeled as single stage service facilities. However, if multiprogramming is allowed, a model with multiservers facilities is necessary.

(5) Service process. For the processing of transactions of the same type, resource requirements in terms of core storage, processing path length, number and distribution of data base access, file structure usage, and memory overlay structure are very similar. The straightforward logic in application programs takes a limited amount of time to execute. Thus, for each type of transaction, the service time distribution may have a small mean and possibly a small variance. However, due to the nature of diversified applications, different transaction types may assume wide variations of service time distributions. It is thus

desirable to have a general distribution for describing the service process.

(6) Scheduling discipline. In view of the nature of different transaction applications, it is generally desirable to use priority scheduling rules in TPS to satisfy different degrees of response performance to different types of transactions.

Thus, TPS can be modeled as multiclass priority queueing systems with Poisson arrivals and general service time distributions. The model queue structure, scheduling discipline, whether an infinite or finite source population, and whether single or multi server(s) depend upon the computer system and the application of the TPS being modeled.

In this dissertation, we will limit our discussion to a single processor, uniprogramming TPS which support a large number of terminals (as is usually true in minicomputer transaction processing systems). We can model this class of TPS as a single server queueing system with multiclass Poisson arrivals and general service time distributions. In Kendall's notation [KEND51], this model is called a multiclass M/G/1 queueing system. (M/G/1 denotes Poisson arrivals, general service time distribution and a single server. The letter M denotes the Markovian property of Poisson arrival.)

1.3 Review and Summary of Priority Scheduling

A scheduling discipline in a queueing system is a decision rule for the service facility to choose among jobs in the queue the next one for service. When we refer to a scheduling discipline as priority scheduling, it is assumed that arriving jobs are grouped into classes based on some external characteristics, and the rule of scheduling depends specifically on the priority class to which a job belongs. The selection decision may or may not depend on other characteristics of the system, such as the queue length, the arrival pattern, the service time required, or service so far rendered.

The following notation will be used throughout this dissertation: For an n-class priority queueing system, we let

λ_i = Poisson arrival rate for class i jobs.

S_i = Random variable for service time of class i jobs.

$E[S_i]$ = Expected service time for class i jobs.

$E[S_i^2]$ = Second moment of service time of class i jobs.

μ_i = 1/E[Si], the service rate for class i jobs.

$\rho_i = \lambda_i/\mu_i$, utilization factor for class i jobs.

E[Wi] = Expected waiting time in queue for class i jobs.

$\lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_n$, total arrival rate.

$\rho = \rho_1 + \rho_2 + \ldots + \rho_n$, total utilization factor.

$\sigma_i = \rho_1 + \rho_2 + \ldots + \rho_i$, partial sum of utilization.

$\mu = \lambda/\rho$, averaged total service rate.

Unless otherwise stated, we will assume $\rho < 1$. We will confine our discussion to scheduling of single server queueing systems with Poisson arrivals and general service time distributions. The class of scheduling that we will study is called "work-conserving", which is defined as the following [KLEI65]:

DEFINITION: A scheduling discipline is work-conserving if it has the following properties:

(i) The processor is always kept busy as long as there are jobs in the system to be processed.

(ii) The total processing time required by any job is not affected by the scheduling decision.

A scheduling is called nonpreemptive if once a job starts its service, it will be served till completion. A scheduling is preemptive if it allows interruption and **gives** service to another job with higher priority. We will limit our discussion to nonpreemptive scheduling rules that are work-conserving.

Below we summarize some results which are relevant to our work. First, we have the following theorem due to Kleinrock [KLEI65] (see also [SCHR70] and [HOAR72]).

THEOREM 1.1. In multiclass M/G/1 queueing systems under all nonpreemptive work-conserving scheduling disciplines, the quantity $\sum_{k=1}^{n} \rho_k E[W_k]$, which is a weighted sum of mean waiting times for all classes, is an invariant, and it is equal to

$$\rho(\sum_{k=1}^{n} \lambda_k E[S_k^2]) \ / \ 2(1-\rho).$$

Now, if we let

$$(1.3.1) \qquad \overline{W} = (\sum_{k=1}^{n} \lambda_k E[S_k^2]) \ / \ 2(1-\rho),$$

then the above theorem becomes:

$$(1.3.2) \qquad \rho_1 E[W_1] + \rho_2 E[W_2] + \ldots + \rho_n E[W_n] = \rho\overline{W}.$$

It is noted that $\overline{W}$ depends only on the first and second moments **of the service time** variable, rather than on the whole service time distribution.

The relation (1.3.2) is generally refered to as the conservation law of mean waiting times. This conservation law puts a linear equality constraint on the set of mean waiting times {E[Wi]}. Any attempt to modify the queueing discipline so as to reduce one of the E[Wi]'s will force an increase in some other E[Wj]. However, this need not be an "even trade" since the weighting factors may be different.

Given n classes of arrivals, if we disregard their priority and serve them strictly on a first-come-first-serve (FCFS) basis, we have the following Pollaczek-Khinchin formula for mean waiting times (See, e.g., [WOLF70]):

THEOREM 1.2. In M/G/1 queueing systems with multiclass arrivals, under FCFS scheduling, the mean waiting times for all classes are all equal to $\overline{W}$. That is, we have

$$E[W_1]_{FCFS} = E[W_2]_{FCFS} = \dots = E[W_n]_{FCFS} = \overline{W}. \tag{1.3.3}$$

We will interchangably use $E[W_{FCFS}]$ and $\overline{W}$ to represent the same quantity as defined in (1.3.1).

Among other priority queueing disciplines, the nonpreemptive head-of-the-line (HOL) scheduling studied by Cobham [COBH54] is the scheduling rule most commonly known. This discipline is also known as fixed priority scheduling or strict priority scheduling. We will call it by the name "fixed priority scheduling."

Under this scheduling rule, jobs are grouped into different priority classes. Upon arrival of a job, it joins the queue for the priority class that it belongs and becomes the last job in the queue. When the service facility is available, the scheduler searches, starting from the highest priority class and moves downward to lower classes, to find the first nonempty queue. The first job from this queue is then taken for service. The scheduling mechanism is illustrated in Figure 1.1. We note that within the same priority class, jobs will be selected on a first-come-first-served (FCFS) basis. The convention of denoting higher priority class with smaller class indices will be followed.

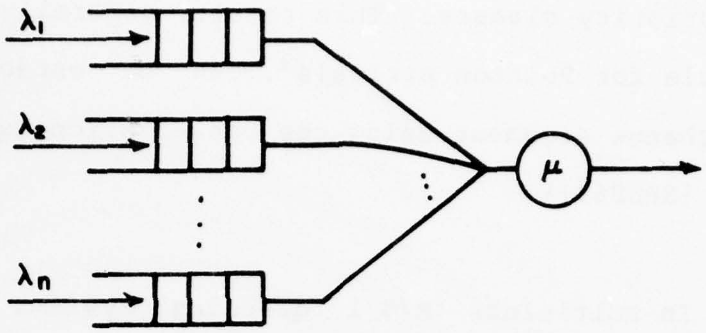


Figure 1.1 Fixed Priority Scheduling Mechanism

The mean waiting time for each priority class under this scheduling rule is given by the following theorem [COBH54]:

THEOREM 1.3. In multiclass M/G/1 queueing systems under fixed priority scheduling, the mean waiting time for each priority class is given by

(1.3.4) $$E[W_i]_{FP} = (1-\rho)\ \overline{W}\ /\ (1-\sigma_{i-1})\ (1-\sigma_i). \qquad (1 \leq i \leq n)$$

As a corollary, we have the following:

(1.3.5) $$(1-\rho)\ \overline{W}\ /\ (1-\rho_1) = E[W_1]_{FP} < \overline{W},$$

and

(1.3.6) $$\overline{W} < E[Wn]_{FP} = \overline{W}\ /\ (1-\rho+\rho_n).$$

The following theorem addresses the problem of optimal ordering of priority classes. This result, generally known as "the μc rule for Poisson arrivals", can be established by an interchange argument using the conservation equation (1.3.2) (See, [BROS63]).

THEOREM 1.4. In multiclass M/G/1 queueing systems under fixed priority scheduling, in order to minimize

$$C = \sum_{k=1}^{n} \lambda_k c_k E[W_k], \tag{1.3.7}$$

priority ordering should be assigned in the descending order of $\mu_i c_i$. That is, higher priorities are given to classes with larger values of $\mu_i c_i$.

In the above theorem, c_i can be thought of as the cost of unit time of waiting per job from class i. In particular, if we let $c_i' = \lambda_i c_i$, then c_i' becomes the total cost of waiting per unit time for class i arrivals. In this case, we have the following corollary.

COROLLARY: In order to minimize

$$C' = \sum_{k=1}^{n} c_k' E[W_k], \tag{1.3.8}$$

priority ordering should be assigned in descending order of c_i'/ρ_i.

Next, we define the rule we call "escalating priority scheduling". This rule was first proposed and studied by Kleinrock [KLEI64].

DEFINITION. In escalating priority scheduling, there is a parameter α_i ($\alpha_i > 0$) associated with each priority class i. When a job from class i arrives to the system, its priority index "escalates" linearly, starting from 0, with rate α_i. Whenever the service facility becomes available, the job with the highest instantaneous priority index is then selected for service. A tie is broken by any arbitration rule.

Theoretically, the arbitration rule for resolving two jobs with the same priority indices is unimportant, since the probability of this event's happening is zero. Practically, however, we can either give preference to the job which has waited longer (i.e., following FCFS rule), or to the job belonging to a higher priority class but has waited for a shorter time (i.e., according to the priorities).

Under this scheduling, the priority function for a job from class i which arrives to the system at time τ_i is,

$$q_i(t) = \alpha_i (t - \tau_i)$$

where t ranges from τ_i until this job obtains service. Thus, priority of each job changes dynamically. The longer a job waits, the higher its priority becomes. Taking waiting time into consideration, this scheduling rule gives priority to jobs that have been in the queue for a long time, even though they may belong to low priority classes.

Figure 1.2 illustrates the priority functions of two jobs from different priority classes which enter the system at two different times. Specifically, at time τ_i, a job from priority class i arrives, and attains priority at a rate equal to α_i. At a later time τ_j, another job from priority class j (j<i) enters the system, and attains its priority at a higher rate α_j. The priority of this job from class j will catch up with that of the job from class i at time t*. Should the server become available any time before t*, the job from class i will obtain service in preference to the job from class j, despite that it is from a "lower" priority class - a class with "lower" priority increasing rate. On the other hand, if the service facility is occupied until after t*, then, when the server is available, the job from class j will be chosen.

We note that under the rule of escalating priority scheduling, preferential treatment is given to classes with larger values of the parameter α_i: For two jobs arriving to the system at the same time, the one with larger value of α_i

gains priority at a faster rate, hence will be served earlier.

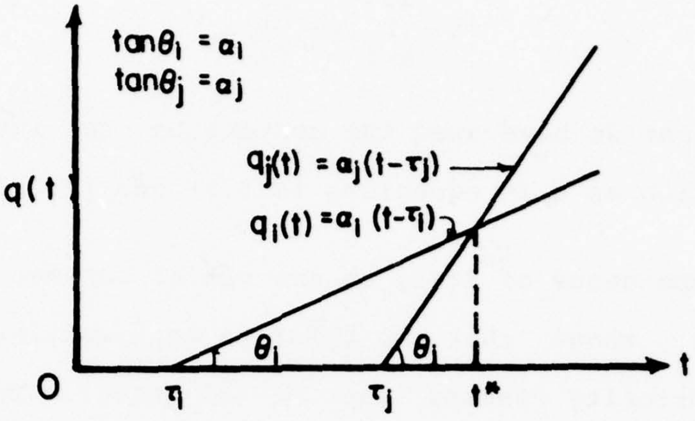


Figure 1.2 Interaction between Priority Indices in Escalating Priority Scheduling

The following theorem, due to Kleinrock [KLEI64], gives the behavior of mean waiting time of each priority class under escalating priority scheduling.

THEOREM 1.5. In multiclass M/G/1 queueing systems under escalating priority scheduling and $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$, the following set of n linear equations hold: For each i, $1 \leq i \leq n$,

$$\left\{1 - \sum_{k=1}^{i-1} \left(1 - \frac{\alpha_i}{\alpha_k}\right) \rho_k\right\} E[Wi] + \sum_{k=i+1}^{n} \left(1 - \frac{\alpha_k}{\alpha_i}\right) \rho_k E[W_k] = \bar{W}. \tag{1.3.9}$$

The mean waiting times for each priority class i can thus be obtained recursively (from n backwards to 1) as

$$(1.3.10) \qquad E[Wi] = \frac{\overline{W} - \sum_{k=i+1}^{n} (1 - \frac{\alpha_k}{\alpha_i}) \rho_k E[W_k]}{1 - \sum_{k=1}^{i-1} (1 - \frac{\alpha_i}{\alpha_k}) \rho_k}$$

Note that we have used the convention to interpret a null summation as 0 in equations (1.3.9) and (1.3.10).

The dependence of E[Wi] on the set of parameters $\{ \alpha_i \}$ in (1.3.10) shows that the relative mean waiting times of different priority classes can be adjusted. This makes escalating priority scheduling more attractive than the conventional fixed priority scheduling, since it is possible to change the values of the parameters for a more desirable system performance.

We remark that since this nonpreemptive scheduling is work-conserving, the conservation law requires a linear constraint (1.3.2) on the performance of mean waiting times of all priority classes.

When Kleinrock first investigated this scheduling rule, he proved the above theorem only for the case when service time distributions are all exponential. In his entire proof of this theorem, only the mean of waiting time distribution is used. Since this mean waiting time is only a function of the first and second moments of the service time distributions, and this is true also for the M/G/1 system as given by equation (1.3.2), the same theorem must also hold for M/G/1 system by using the values of W given by (1.3.2). When Kleinrock summarized this result later in his book [KLEI76], he seemed to imply this, although he did not clearly make the point. (He had kept the sentence "we study this system for the case of exponential service times ..." in his book.)

1.4 Problem Definition and Overview

As we stated earlier, in transaction processing systems, priority scheduling is required in order to give different response times to different types of transactions. In many existing TPS, fixed priority scheduling is implemented. When these systems are operating under heavy congestion, many low priority transactions suffer excessively long response times. However, it is possible that some high priority transactions are still obtaining unnecessarily fast responses. This suggests that a scheduling rule other than fixed priority scheduling is desirable.

Studies of human behavior in man-machine interactions have shown that long response times will interrupt the continuity of human thinking, and will drasically reduce the productivity of the terminal users (See, [MILL68] and [CARB68]). In general, it is considered that for response times, the faster, the better. However, we all know that there is also a human reaction limit. A response time that is faster than what a human being can react is not necessary. Figure 1.3 shows the situation of response times under fixed priority scheduling, together with a desirable performance that lies within the two human response limits.

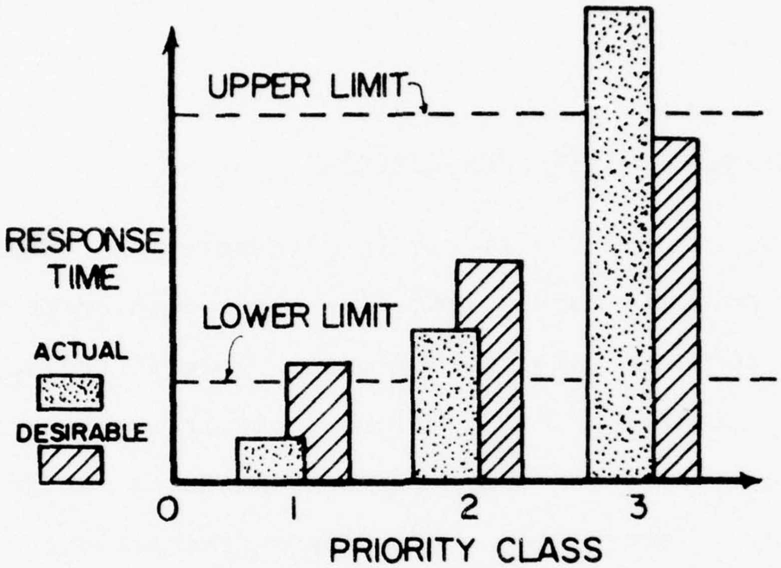


Figure 1.3 Response Times Behavior under Fixed Priority Scheduling Rule

The above discussion suggests that it is desirable to have a scheduling rule to provide reasonable response times to low priority transactions so that the continuity of human thinking will not be interrupted. The escalating priority scheduling rule appears to be a good scheduling rule to consider because it can discriminate against high priority transactions in not providing faster response time than necessary and thus allow some of the TPS resources to respond to lower priority transactions.

In analyzing queueing models, it is generally desirable to obtain measures other than just mean waiting times. For analyzing multiclass M/G/1 queueing models, in order to obtain the waiting time distributions, specific service times distributions must generally be used and even then, it is usually difficult to obtain analytic results. In studying escalating priority scheduling for TPS, we will limit our investigations to the mean and variance of waiting times.

Some of the questions that need to be answered in determining if escalating priority scheduling is indeed applicable to transaction processing systems are:

(1) What is the feasible performance space of mean waiting times under this scheduling? What are the bounds of mean waiting time for each priority class?

(2) How do we find the values of the control parameters that will optimize different cost objective functions?

We will answer these questions, plus investigating some other **aspects** of escalating priority scheduling, in this dissertation. In Chapter 2, we investigate the limitations of this scheduling, derive the feasible performance space of mean waiting times and show that the range of this scheduling rule is from fixed priority scheduling to first-come-first-serve scheduling with no priorities.

In Chapter 3, we study the problem of optimization for several cost functions. Our solution procedure is divided into two stages: First, we determine a set of optimal mean waiting times for a given cost function. Then, we determine the values of control parameters for this set of mean waiting times.

Chapter 4 discusses several other aspects of escalating priority scheduling, such as bounds on the ratios of mean waiting times, the behavior of these ratios of waiting times under saturated congestion and variances of waiting times under different levels of discrimination.

In Chapter 5, we compare the behavior of escalating priority scheduling with other "adjustable priority scheduling rules", discuss the use of escalating priority scheduling in transaction processing systems, and give conclusions and directions for future research.

CHAPTER 2

ESCALATING PRIORITY SCHEDULING DISCIPLINE

2.0 Introduction

We have seen in Chapter 1 that escalating priority scheduling has a set of control parameters. By adjusting the settings of these parameters, different mean waiting times can be achieved. In this chapter, we study the limitations of this scheduling, derive the feasible performance space of mean waiting times for each class of jobs, and develop an algorithm for determining values of the control parameters given a set of feasible expected waiting times.

In Section 2.1, we introduce a general concept of scheduling rules with adjustable parameters. For this class of scheduling rules, we derive a set of constraints (bounds) on the mean waiting times of jobs from different priority classes and show that they can cover the spectrum of scheduling from no discrimination to maximum discrimination among priority classes by adjusting their parameters.

In Section 2.2, we show that the set of constraints derived in Section 2.1 is sufficient for characterizing the feasible performance space of mean waiting times under escalating priority scheduling. That is, any combination of mean waiting times satisfying this set of constraints can be achieved by escalating priority scheduling. We then develop an algorithm for determining the values of the control parameters given any set of feasible mean waiting times.

In section 2.3, we discuss interpretation of boundary points of the feasible performance space under escalating priority scheduling, and show that non-extreme points on the boundary lead to a natural definition of a mixed rule of escalating priority and fixed priority scheduling.

2.1 Adjustable Priority Scheduling Rules

In multiclass queueing systems, priority scheduling is used to discriminate among different classes of arrivals. Some of the scheduling rules are furnished with a set of parameters that can be adjusted to provide different discriminations among the different classes of jobs. We call this class of scheduling rules the "adjustable priority scheduling rules", and define it formally as the following:

DEFINITION. In an adjustable priority scheduling rule, there is a set of parameters that can be varied to enforce different degrees of discriminations among priority classes.

In adjustable priority scheduling rules, the priority index of a job must be a function of more than the parameter associated with the priority class, otherwise the scheduling rule will become fixed priority scheduling without any adjustability. Usually, the priority index function depends, in addition to the parameter associated with the priority class, on the history or the current state of the system. For example, in escalating priority scheduling, the priority index of a job depends on both the waiting time and the parameter associated with the job. Other examples include Earliest Due Date scheduling and deescalating priority scheduling (these scheduling rules are discussed in Chapter 5.)

In the following, we investigate the limitations of mean waiting times under any adjustable priority scheduling rule. Please recall that we have restricted ourselves to scheduling rules that are both work-conserving and nonpreemptive. First, we have the following theorem.

THEOREM 2.1. In multiclass M/G/1 queueing systems, under any nonpreemptive adjustable priority scheduling rule, the mean waiting times for each class is bounded by:

$$(2.1.1) \qquad (1-\rho)\overline{W}/(1-\rho_i) \leq E[Wi] \leq \overline{W}/(1-\rho+\rho_i). \quad (1\leq i\leq n)$$

<u>Proof</u>: For each priority class i, the maximum preference that can be given to this class under any adjustable priority scheduling rule is when jobs from this class are given utmost preference over any other classes' jobs. Under this condition, no job from any other class will ever be selected

for service as long as there are jobs from class i present in the system. But this is exactly what happens when the system is operated under fixed priority scheduling that gives class i the highest prioity. Thus, the mean waiting time of class i when it is set as the highest priority class under fixed priority scheduling gives the lower bound of E[Wi] for any adjustable priority scheduling rule. From (1.3.5) of Theorem 1.3, we have the first half of inequality set (2.1.1).

Similarly, the maximum discrimination that can be applied against class i under any adjustable priority scheduling rule is when jobs from this class are given least preference. As long as there are jobs from other classes present in the system, no job from class i will be selected for service. This is what happens in fixed priority scheduling, when class i is set as the lowest priority class. Therefore, the upper bound of E[Wi] can be obtained from (1.3.5) of Theorem 1.3, which is the second half of inequality set (2.1.1).

In the following, we show that not only is the mean waiting time of each priority class bounded, but a weighted sum of the mean waiting times for any number of classes is also bounded.

THEOREM 2.2. In multiclass M/G/1 queueing systems under any nonpreemptive adjustable priority scheduling rule, for any set I of priority class indices, we have the following bounds on a weighted sum of mean waiting times for jobs with class indices in this set:

$$(2.1.2)\quad (1-\rho)(\sum_{k\in I}\rho_k)\bar{W}/(1-\sum_{k\in I}\rho_k) \le \sum_{k\in I}\rho_k E[W_k] \le (\sum_{k\in I}\rho_k)\bar{W}/(1-\rho+\sum_{k\in I}\rho_k)\ .$$

In the above expression, the notation $\sum_{k\in I}$ denotes the summation for all classes with class index in I. We will exclude the case when I is equal to the set of all priority indices, because in this case, (2.1.2) becomes the conservation constraint (1.3.2).

<u>Proof</u>: For any given I as the set of class indices of an arbitrary group of priority classes, let the mean waiting time of all jobs in this group be $E[W_I]$. Let N denote the set of all class indices. That is, N = {1, 2, ..., n}. Let N-I denote the set of class indices which are not in I. Then if we look at the classes with class indices in I as distinct classes, from the conservation law we have

$$\sum_{k\in I}\rho_k E[W_k] + \sum_{k\in N-I}\rho_k E[W_k] = \rho\bar{W}.$$

Now, if we consider the whole group of priority classes with class indices in I as a single class, its congestion factor is then equal to $\sum_{k\in I}\rho_k$. From this point of view, the conservation equation becomes

$$\left(\sum_{k\in I} \rho_k\right) E[W_I] + \sum_{k\in N-I} \rho_k E[W_k] = \rho \bar{W}.$$

From the above two equations, we have

$$E[W_I] = \sum_{k\in I} \rho_k E[W_k] \Big/ \sum_{k\in I} \rho_k . \qquad (2.1.3)$$

Now, for the mean waiting time $E[W_I]$ of all jobs in this "combined class", under any adjustable priority scheduling rule, from Theorem 2.1, we have

$$(1-\rho)\bar{W}/\left(1-\sum_{k\in I} \rho_k\right) \leq E[W_I] \leq \bar{W}/\left(1-\rho+\sum_{k\in I} \rho_k\right).$$

Substituting (2.1.3) into the above expression, and multiplying by the quantity $\left(\sum_{k\in I} \rho_k\right)$ throughout, we obtain (2.1.2).

We remark that in Theorem 2.2, for the special case when the set I consists of only one element (e.g., I={i}), then (2.1.2) becomes

$$(1-\rho)\rho_i\bar{W}/(1-\rho_i) \leq \rho_i E[Wi] \leq \rho_i \bar{W}/(1-\rho+\rho_i).$$

Cancelling out the factor ρ_i, this becomes an inequality of (2.1.1). Thus, if we allow I to be any set of class indices, then (2.1.2) includes (2.1.1) as a special case.

We will now examine the two sets of inequalities in (2.1.2), and show that when the conservation equation (1.3.2) holds, these two sets of inequalities can be derived from each other:

From the first set of inequalities, for each I, we have

$$\sum_{k\in I} \rho_k E[W_k] \geq (1-\rho)\overline{W}\Big(\sum_{k\in I} \rho_k\Big)/\Big(1-\sum_{k\in I} \rho_k\Big).$$

From the conservation law, we have

$$(1.3.2) \qquad \rho_1 E[W_1] + \rho_2 E[W_2] + \ldots + \rho_n E[W_n] = \rho\,\overline{W},$$

Taking the difference of the above two equations, we then have

$$\sum_{k\in N-I} \rho_k E[W_k] \leq \rho\overline{W} - (1-\rho)\overline{W}\Big(\sum_{k\in I} \rho_k\Big)/\Big(1-\sum_{k\in I} \rho_k\Big)$$

$$= \Big\{\rho\overline{W} - \Big(\sum_{k\in N-I} \rho_k\Big)\rho\overline{W} - \Big(\sum_{k\in I} \rho_k\Big)\overline{W} + \Big(\sum_{k\in I} \rho_k\Big)\rho\overline{W}\Big\}/\Big(1-\sum_{k\in I} \rho_k\Big)$$

$$= \Big(\sum_{k\in N-I} \rho_k\Big)\overline{W}/\Big(1-\rho+\sum_{k\in N-I} \rho_k\Big).$$

This is an inequality in the second set of (2.1.2). Since this is true for all i, all inequalities in the second set of (2.1.2) can be derived from the first set of inequalities.

Similarly, if we take the difference of (1.3.2) with any inequality in the second half of (2.1.2), we obtain an inequality in the first half of (2.1.2). This holds true for all i.

Since the conservation law holds for any nonpreemptive adjustable priority scheduling, only one set of inequalities in (2.1.2) is needed to characterize the mean waiting times behavior.

Summarizing the above discussion, we have the following necessary conditions for any nonpreemptive adjustable priority scheduling rule:

$$(1.3.2) \qquad \rho_1 E[W_1] + \rho_2 E[W_2] + \ldots + \rho_n E[W_n] = \rho \overline{W},$$

and

$$(2.1.4) \qquad \forall I \subset N \quad \sum_{k \in I} \rho_k E[W_k] \geq (1-\rho)\Big(\sum_{k \in I} \rho_k\Big)\overline{W}/\Big(1-\sum_{k \in I} \rho_k\Big).$$

In the above expression, we have used the notation $\forall I \subset N$ to mean "for each proper subset I of N", where a proper subset of N is defined to be a nonempty subset of N that is not equal to N itself [LIPS64]. Thus, for example, when N = {1, 2, 3}, then I can be any of the following: {1}, {2}, {3}, {1, 2}, {1, 3} and {2, 3}. Therefore, when we require I to run through all the possible subsets of N, (2.1.4) becomes a set of six inequalities. We remark that for a set N of n elements, (2.1.4) consists of 2^n-2 inequalities.

The above summary states that for any setting of an adjustable priority scheduling rule, the point W = $(E[W_1], E[W_2], \ldots, E[W_n])$ must lie within the space defined by (1.3.2) and (2.1.4). In other words, the space defined by (1.3.2) and (2.1.4) is the maximum possible performance space of mean waiting times under any adjustable priority scheduling rule.

We now give the geometrical representations of these constraints in two and three dimensional spaces.

Figure 2.1 shows the bounds of mean waiting times in a two-class queueing system under any adjustable priority scheduling rule. The conservation equation (1.3.2) in the two-dimensional space is a straight line L: $\rho_1 E[W_1] + \rho_2 E[W_2] = \rho\overline{W}$. The set of constraints (2.1.4) consists of two inequalities:

$$\text{(a)}\quad \rho_1 E[W_1] \geq (1-\rho)\rho_1\overline{W}/(1-\rho_1);$$
$$\text{and}\quad \text{(b)}\quad \rho_2 E[W_2] \geq (1-\rho)\rho_2\overline{W}/(1-\rho_2).$$

Let L_1: $\rho_1 E[W_1] = (1-\rho)\rho_1\overline{W}/(1-\rho_1)$ and L_2: $\rho_2 E[W_2] = (1-\rho)\rho_2\overline{W}/(1-\rho_2)$ correspond to the boundaries of the above two inequalities. Then we see that L_1 and L intercept at $Q_{12} = ((1-\rho)\overline{W}/(1-\rho_1), \overline{W}/(1-\rho_1))$; while L_2 and L intercept at $Q_{21} = (\overline{W}/(1-\rho_2), (1-\rho)\overline{W}/(1-\rho_2))$.

Thus, the maximum performance space of mean waiting times is the line segment $\overline{Q_{12}Q_{21}}$. It is interesting to note that point $P = (\overline{W}, \overline{W})$ lies on the line $\overline{Q_{12}Q_{21}}$. This point corresponds to a scheduling rule which applies no discrimination among the two priority classes (e.g., FCFS or last-come-first-serve (LCFS) scheduling).

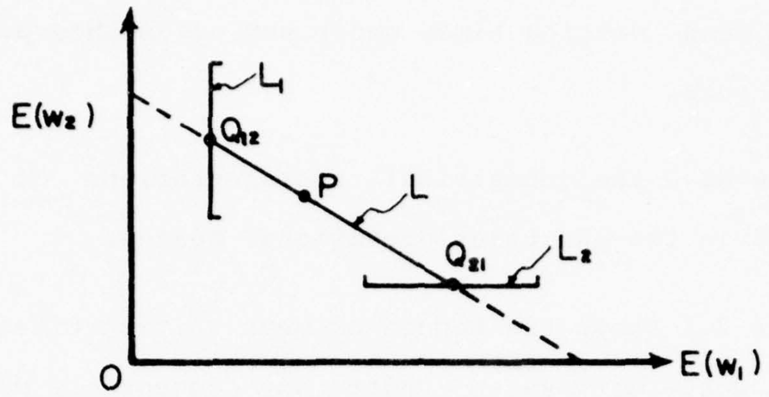


Figure 2.1 Bounds of Mean Waiting Times in a Two-Class Queueing Model

We note that for points in between P and Q_{12}, we have $E[W_1] < E[W_2]$. Thus, they correspond to scheduling rules that give preference to class 1. Similarly, points lying between P and Q_{21} correspond to scheduling rules that give preference to class 2. The extreme point Q_{12} corresponds to the mean waiting times performance of fixed priority scheduling which gives high priority to class 1. Similarly, point Q_{21} corresponds to mean waiting times performance of fixed priority scheduling with reversed ordering of priorities.

Next, we will discuss 3-class queueing systems. In 3-dimensional space, the conservation equation (1.3.2) becomes a plane S: $\rho_1 E[W_1] + \rho_2 E[W_2] + \rho_n E[W_n] = \rho \overline{W}$. It is shown as the plane passing through the points X1, X2, X3 in Figure 2.2.

In this case, constraints (2.1.4) is the following set of 2^3-2 or 6 inequalities:

(1) $\rho_1 E[W_1] \geq (1-\rho)\rho_1\overline{W}/(1-\rho_1)$

(2) $\rho_2 E[W_2] \geq (1-\rho)\rho_2\overline{W}/(1-\rho_2)$

(3) $\rho_3 E[W_3] \geq (1-\rho)\rho_3\overline{W}/(1-\rho_3)$

(4) $\rho_1 E[W_1] + \rho_2 E[W_2] \geq (1-\rho)(\rho_1+\rho_2)\overline{W}/(1-\rho_1-\rho_2)$

(5) $\rho_1 E[W_1] + \rho_3 E[W_3] \geq (1-\rho)(\rho_1+\rho_3)\overline{W}/(1-\rho_1-\rho_3)$

(6) $\rho_2 E[W_2] + \rho_3 E[W_3] \geq (1-\rho)(\rho_2+\rho_3)\overline{W}/(1-\rho_2-\rho_3)$

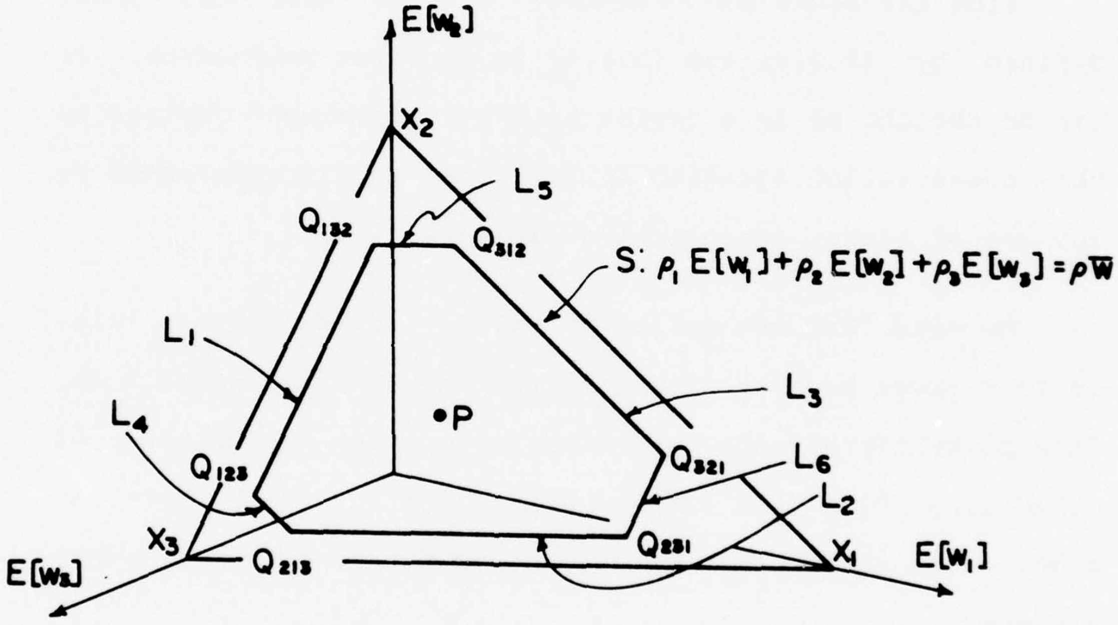


Figure 2.2 Bounds of Mean Waiting Times in a Three-Class Queueing Model

The six lines corresponding to the boundaries of the above six inequalities are shown as L_1 through L_6 in Figure 2.2. The six vertices Q_{123}, Q_{213}, Q_{231}, Q_{321}, Q_{312} and Q_{132} correspond to the mean waiting times performances of the six different orderings under fixed priority scheduling. The subscript of each point represents the ordering of priority classes. Thus, for example, Q_{213} represents the mean waiting times performance of fixed priority scheduling when class 2 is given highest priority, class 1 the next, and class 3 the lowest priority. Once again, the point $P = (\overline{W}, \overline{W}, \overline{W})$ on the plane corresponds to scheduling with no discriminations among the three priority classes.

From the above two examples, we see that the space defined by (1.3.2) and (2.1.4) is a convex polyhedron. It can be thought of as a region on the "hyperplane" defined by the conservation equation (1.3.2) that is circumferenced by the set of linear inequalities (2.1.4).

We note that the point $P = (\overline{W}, \overline{W}, \ldots, \overline{W})$ always lies in this space because it satisfies both (1.3.2) and (2.1.4). This point corresponds to mean waiting times performance of scheduling rules that have no discrimination (FCFS, LCFS, or other rules that do not apply discrimination among priority classes).

We also note that from the proof of Theorem 2.1, the bounds of adjustable priority schedulings are obtained from the mean waiting times of fixed priority scheduling, which can be achieved by using either FCFS or LCFS to determine the ordering of jobs within the same class.

Therefore, this class of adjustable priority scheduling rules covers the spectrum of scheduling rules which do not distinguish among priority classes to those that separate priority classes to the maximum extent.

2.2 Feasible Performance Space of Mean Waiting Times

In Section 2.1, we showed that mean waiting times performance of any adjustable priority scheduling must satisfy the following constraints:

$$(1.3.2) \qquad \rho_1 E[W_1] + \rho_2 E[W_2] + \ldots + \rho_n E[W_n] = \rho\overline{W},$$

and

$$(2.1.4) \qquad \forall I \subset N \quad \sum_{k \in I} \rho_k E[W_k] \geq (1-\rho)(\sum_{k \in I} \rho_k)\overline{W}/(1-\sum_{k \in I} \rho_k).$$

Since these constraints are linear, the region bounded by these constraints is a convex polyhedron. The interior points of this space are defined by (1.3.2) and

$$(2.2.1) \qquad \forall I \subset N \quad \sum_{k \in I} \rho_k E[W_k] > (1-\rho)(\sum_{k \in I} \rho_k)\overline{W}/(1-\sum_{k \in I} \rho_k),$$

where (2.2.1) is obtained by replacing "$\geq$" with "$>$" in (2.1.4).

In this section, we show that we can find a set of values for the control parameters $\{\alpha_i\}$ in escalating priority scheduling for any set of mean waiting times satisfying (1.3.2) and (2.2.1). Thus, (1.3.2) and (2.2.1) characterize the feasible performance space of mean waiting times under escalating priority scheduling. In addition, we present an algorithm for determining the values of $\{\alpha_i\}$ given a set of feasible mean waiting times.

Before we proceed to derive the procedure for determining values of the control parameters $\{\alpha_i\}$, we examine Theorem 1.5 more thoroughly. We recall that in Theorem 1.5, equations (1.3.9) and (1.3.10) were obtained under the condition that $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n$. This corresponds to the setting of orderings that gives class 1 the highest priority, class 2 the second, and so on, and class n the lowest priority. Under this setting of priorities, we have the following:

THEOREM 2.3. In multiclass M/G/1 queueing systems under escalating priority scheduling, if we are given $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n$, then

$$E[W_1] \leq E[W_2] \leq \dots \leq E[W_n]. \qquad (2.2.2)$$

Proof: We shall show that

$$E[W_i] \leq E[W_{i+1}] \quad \text{for all } i, 1 \leq i \leq n-1 .$$

From (1.3.9), we have

$$E[W_i] = \frac{\overline{W} - \sum_{k=i+1}^{n} (1 - \frac{\alpha_k}{\alpha_i}) \rho_k E[W_k]}{1 - \sum_{k=1}^{i-1} (1 - \frac{\alpha_k}{\alpha_i}) \rho_k}$$

$$\leq \frac{\overline{W} - \sum_{k=i+2}^{n} (1 - \frac{\alpha_k}{\alpha_i}) \rho_k E[W_k]}{1 - \sum_{k=1}^{i-1} (1 - \frac{\alpha_i}{\alpha_k}) \rho_k}$$

$$\leq \frac{\overline{W} - \sum_{k=i+2}^{n} (1 - \frac{\alpha_k}{\alpha_{i+1}}) \rho_k E[W_k]}{1 - \sum_{k=1}^{i-1} (1 - \frac{\alpha_{i+1}}{\alpha_k}) \rho_k}$$

$$\leq \frac{\overline{W} - \sum_{k=i+2}^{n} (1 - \frac{\alpha_k}{\alpha_{i+1}}) \rho_k E[W_k]}{1 - \sum_{k=1}^{i} (1 - \frac{\alpha_{i+1}}{\alpha_k}) \rho_k}$$

$$= E[W_{i+1}] \; .$$

Thus, (2.2.2) holds.

In the following theorem, we describe an algorithm for determining the values of $\{\alpha_i\}$ given a set of mean waiting times $(E[W_1], E[W_2], \ldots, E[W_n])$ satisfying (1.3.2), (2.2.1) ana (2.2.2).

THEOREM 2.4. In a multiclass M/G/1 queueing system with $\rho < 1$, a set of values for $\{\alpha_i\}$ can be obtained for escalating priority scheduling to achieve any given set of mean waiting times $(E[W_1], E[W_2], \ldots, E[W_n])$ that satisfies (1.3.2), (2.2.1) and (2.2.2). Define, recursively (for i from n-1 backwards to 1)

(2.2.3) $$a_i = (r_i a_{i+1} + \rho_i)E[W_i]/E[W_{i-1}],$$

(2.2.4) $$b_i = (\overline{W} - \sum_{k=i}^{n} \rho_k E[W_k])/E[W_{i-1}] - (1-\sigma_{i-1}),$$

(2.2.5) $$c_i = \rho_i - b_{i+1} - r_i a_{i+1},$$

where r_i is the unique positive root of

(2.2.6) $$a_{i+1}X^2 + b_{i+1}X + c_{i+1} = 0$$

and, initially,

(2.2.7) $$a_n = \rho_n E[W_n]/E[W_{n-1}],$$

(2.2.8) $$b_n = (\overline{W} - \rho_n E[W_n])/E[W_{n-1}] - (1-\sigma_{n-1}),$$

(2.2.9) $$c_n = (1-\sigma_{n-1}) - \overline{W}/E[W_n].$$

Then, $\{\alpha_i\}$ can be obtained as:

(2.2.10) $$\alpha_i = 1 / \prod_{k=i}^{n-1} r_k \qquad i = 1, 2, \ldots, n.$$

Note that in (2.2.10), we have used the convention to interpret a null product as 1 for the case when i=n. Also. r_i is defined only for $1 \leq i \leq n-1$. From (2.2.10) we have $r_i = \alpha_{i+1}/\alpha_i$. Since we are given that $E[W_i] \leq E[W_{i+1}]$, therefore $\alpha_i \geq \alpha_{i+1}$. thus, each r_i will be smaller than or equal to 1.

We remark that since the conservation law puts a linear constraint on the performance of mean waiting times for each class of jobs, there are only n-1 degrees of freedom to specify a desirable mean waiting times performance. For these degrees of freedom, only n-1 independent variables are needed. Therefore, for the set of n control parameters, there is one degree of redundancy and only n-1 degrees of adjustability can be achieved.

<u>Proof</u>: Under escalating priority scheduling, for any set of control parameters $\{\alpha_i\}$, the mean waiting times $\{E[W_i]\}$ must satisfy the following set of n equations (from Theorem 1.5):

$$(1.3.9) \qquad \{1 - \sum_{k=1}^{i-1} (1 - \frac{\alpha_i}{\alpha_k}) \rho_k\} E[W_i] + \sum_{k=i+1}^{n} (1 - \frac{\alpha_k}{\alpha_i}) \rho_k E[W_k] = \overline{W} .$$

Our problem now is to determine the values of $\{\alpha_i\}$ given $E[W_i]$, $1 \leq i \leq n$, which satisfy (1.3.9) above. We will first solve this set of equations for $\{\alpha_i\}$.

Let us define, for each i, $1 \leq i \leq n-1$,

$$(2.2.11) \qquad r_i = \alpha_{i+1}/\alpha_i .$$

Then, we see that for any k > i,

$$\frac{\alpha_k}{\alpha_i} = \frac{\alpha_{i+1}}{\alpha_i} \cdot \frac{\alpha_{i+2}}{\alpha_{i+1}} \cdot \ldots \cdot \frac{\alpha_k}{\alpha_{k-1}} = \prod_{k=i}^{k-1} r_j .$$

Similarly, for k < i,

$$\frac{\alpha_i}{\alpha_k} = \frac{\alpha_{k+1}}{\alpha_k} \cdot \frac{\alpha_{k+2}}{\alpha_{k+1}} \cdot \ldots \cdot \frac{\alpha_i}{\alpha_{i-1}} = \prod_{j=k}^{i-1} r_j .$$

Thus, (1.3.9) can be rewritten as

$$\{1 - \sum_{k=1}^{i-1} (1 - \prod_{j=k}^{i-1} r_j)\rho_k\}E[W_k] + \sum_{k=i+1}^{n} (1 - \prod_{j=i}^{k-1} r_j)\rho_k E[W_k] = \overline{W} .$$

Shifting those terms which do not contain the factor r_j to the right of the equality sign and dividing by E[Wi], we obtain

$$(2.2.12) \qquad \sum_{k=1}^{i-1} \rho_k (\prod_{j=k}^{i-1} r_j) - \sum_{k=i+1}^{n} (\rho_k E[W_k]/E[W_i])(\prod_{j=i}^{k-1} r_j)$$

$$= (\overline{W} - \sum_{k=i+1}^{n} \rho_k E[W_k])/E[W_i] - (1 - \sigma_{i-1}) .$$

Equation (2.2.12) is a set of n nonlinear equations of the n-1 variables $\{r_i : 1 \le i \le n-1\}$ which is of degree n-1. For later reference, the i-th equation of (2.2.12) will be defined as (2.2.12.i). We will now solve this set of simultaneous equations for r_i, $1 \le i \le n-1$.

It is important to note that if these r_i's can be successfully obtained, then we can use (2.2.11) to obtain $\{\alpha_i\}$:

$$\alpha_i = \frac{\alpha_{i+1}}{r_i} = \frac{1}{r_i} \cdot \frac{\alpha_{i+2}}{r_{i+1}} = \ldots$$
$$= \alpha_n / (\prod_{k=i}^{n-1} r_k) .$$

For the extra degree of freedom, we arbitrarily let

$$\alpha_n = 1.$$

Thus, we have

$$(2,2,10) \qquad \alpha_i = 1 / \prod_{k=i}^{n-1} r_k ,$$

which is exactly what was stated in the theorem.

To solve for $\{r_i : 1 \le i \le n-1\}$, we first multiply (2.2.12.n-1) by $-r_{n-1}$ to obtain

$$(2.2.13) \qquad - \sum_{k=1}^{n-2} \rho_k (\prod_{j=k}^{n-3} r_j) r_{n-1} + (\rho_n E[W_n] / E[W_{n-1}]) r_{n-1}^2$$
$$= \{ (\overline{W} - \rho_n E[W_n]) / E[W_{n-1}] - (1 - \sigma_{n-2}) \} r_{n-1} .$$

Equation (2.2.12.n) when written out, becomes

$$(2.2.12.n) \qquad \sum_{k=1}^{n-1} \rho_k (\prod_{j=k}^{n-1} r_j) = \overline{W} / E[W_n] - (1 - \sigma_{n-1})$$

or,

$$\sum_{k=1}^{n-2} \rho_k (\prod_{j=k}^{n-2} r_j) r_{n-1} + \rho_{n-1} r_{n-1} = \overline{W} / E[W_n] - (1 - \sigma_{n-1}) .$$

Adding the above equation to (2.2.13), we obtain

$$(\rho_n E[W_n] / E[W_{n-1}]) r_{n-1}^2 + \{ (\overline{W} - \rho_n E[W_n]) / E[W_{n-1}] - (1-\sigma_{n-1}) \} r_{n-1}$$

$$+ (1 - \sigma_{n-1} - \overline{W} / E[W_n]) = 0 .$$

This is a quadratic equation of the variable r_{n-1} only. We note that the coefficients of this quadratic equation are the a_n, b_n and c_n defined in (2.2.7) through (2.2.9) respectively. Thus, r_{n-1} is a root of

(2.2.6.n) $\quad a_n X^2 + b_n X + c_n = 0.$

Since r_{n-1} is defined to be the ratio of α_n and α_{n-1}, and, since all the α_i's are positive, we should expect r_{n-1} to be positive. Also, in order that r_{n-1} be uniquely defined, we need to show that there is only one positive solution of (2.2.6.n).

This is the case if we can show that $a_n c_n < 0$. This is because if x_1 and x_2 are two roots of $a_n X^2 + b_n X + c_n = 0$, then $x_1 \cdot x_2 = c_n/a_n$. Thus, $a_n c_n < 0$ if and only if $c_n/a_n < 0$, which means that x_1 and x_2 are opposite in sign. This guarantees that either x_1 or x_2 is positive.

Now, from (2.2.7), it is easily seen that $a_n > 0$. Therefore, we need to show that $c_n < 0$. From (2.2.1) with $I = \{1, 2, \ldots, n-1\}$, we have

$$\sum_{k=1}^{n-1} \rho_k E[W_k] > (1-\rho)\overline{W}\sigma_{n-1} / (1-\sigma_{n-1}) .$$

Taking the difference of (1.3.2) with the above inequality, we obtain

$$\rho_n E[W_n] < \rho\overline{W} - (1-\rho)\overline{W}\sigma_{n-1}/(1-\sigma_{n-1})$$

$$= (\rho\overline{W}-\sigma_{n-1}\rho\overline{W}-\sigma_{n-1}\overline{W}+\sigma_{n-1}\rho\overline{W}) / (1-\sigma_{n-1})$$

$$= (\rho\overline{W}-\sigma_{n-1}\overline{W}) / (1-\sigma_{n-1})$$

$$= \rho_n\overline{W} / (1-\sigma_{n-1}) .$$

Thus, $\quad E[W_n] < \overline{W}/(1-\sigma_{n-1})$

or, $\quad 1-\sigma_{n-1} < \overline{W}/E[W_n],$

so $\quad c_n = (1-\sigma_{n-1}) - \overline{W}/E[W_n] < 0.$

We will now use the principle of induction to show the following:

(a) Once we found r_k, $i \le k \le n-1$, then r_{i-1} must be a root of the quadratic equation $a_i X^2 + b_i X + c_i = 0$ where a_i, b_i and c_i are defined by (2.2.3) through (2.2.5).

(b) There is only one positive root to the above quadratic equation $a_i X^2 + b_i X + c_i = 0$.

First we show (a): From (2.2.12), we get

$$\text{(2.2.12.i-1)} \quad \sum_{k=1}^{i-2} \rho_k \Big(\prod_{j=k}^{i-2} r_j\Big) - \sum_{k=1}^{n} (\rho_k E[W_k]/E[W_{i-1}])\Big(\prod_{j=i-1}^{k} r_j\Big)$$

$$= -\Big\{\Big(\overline{W} - \sum_{k=i}^{n} \rho_k E[W_k]\Big)/E[W_{i-1}] - (1-\sigma_{i-2})\Big\} r_{i-1} .$$

Multiply it by $-r_{i-1}$, we get

$$-\sum_{k=1}^{i-2}\rho_k(\prod_{j=k}^{i-1}r_j)+\sum_{k=1}^{i}(\rho_k E[W_k]/E[W_{i-1}])(\prod_{j=i-1}^{k}r_j)$$

$$=-\{(\bar{W}-\sum_{k=1}^{n}\rho_k E[W_k])/E[W_{i-1}]-(1-\sigma_{n-2})\}r_{i-1}\,.$$

Now, add the above equation to (2.2.12), we have

$$\rho_{i-1}r_{i-1}+\sum_{k=i}^{n}(\rho_k E[W_k]/E[W_{i-1}])(\prod_{j=i}^{k-1}r_j)r_{i-1}^2$$

$$-\sum_{k=i+1}^{n}(\rho_k E[W_k]/E[W_{i-1}])(\prod_{j=i}^{k}r_j)$$

$$=-\{(\bar{W}-\sum_{k=i}^{n}\rho_k E[W_k])/E[W_{i-1}]-(1-\sigma_{i-2})r_{i-1}$$

$$+(\bar{W}-\sum_{k=i+1}^{n}\rho_k E[W_k])/E[W_i]-(1-\sigma_{i-1})\,.$$

After shifting and combining terms, we obtain

$$\sum_{k=i}^{n}(\rho_k E[W_k]/E[W_{i-1}])(\prod_{j=i}^{k-1}r_j)r_{i-1}^2$$

$$+\{(\bar{W}-\sum_{k=i}^{n}\rho_k E[W_k])/E[W_{i-1}]-(1-\sigma_{i-1})\}r_{i-1}$$

$$-\sum_{k=i+1}^{n}(\rho_k E[W_k]/E[W_i])(\prod_{j=i}^{k-1}r_j)$$

$$-(\bar{W}-\sum_{k=i+1}^{n}\rho_k E[W_k])/E[W_{i-1}]+(1-\sigma_{i-1})=0\,.$$

This is an equation of r_{i-1} only, since all the r_k's for $k \geq i$ are known. Let the coefficients of the above quadratic equation be a_i', b_i' and c_i' respectively. We then need to show that $a_i' = a_i$, $b_i' = b_i$ and $c_i' = c_i$ as defined in (2.2.3) through (2.2.5).

Comparing b_i' with b_i as defined by (2.2.4), we can see that b_i' is exactly b_i. So we will show (i) $a_i' = a_i$ and (ii) $c_i' = c_i$.

$$\text{(i) } a_i' = \sum_{k=i}^{n} (\rho_k E[W_k]/E[W_{i-1}])(\prod_{j=i}^{k} r_j)$$

$$= \rho_i E[W_i]/E[W_{i-1}] + \sum_{k=i+1}^{n} (\rho_k E[W_k]/E[W_{i-1}])(\prod_{j=i}^{k} r_j)$$

$$= (E[W_i]/E[W_{i-1}])\{\rho_i + \sum_{k=i+1}^{n} (\rho_k E[W_k]/E[W_i])(\prod_{j=i+1}^{k} r_j) r_i$$

$$= (E[W_i]/E[W_{i-1}])(\rho_i + a'_{i+1} r_i) .$$

But, initially we know that $a_n' = a_n$. By the principle of induction, we have $a_i' = a_i$.

$$\text{(ii) From (2.2.3), } a_{i+1} = \sum_{k=i+1}^{n} (\rho_k E[W_k]/E[W_i])(\prod_{j=i+1}^{k-1} r_j) .$$

$$\text{Thus, } c_i = - a_{i+1} r_i - \{(\overline{W} - \sum_{k=i+1}^{n} \rho_k E[W_k]/E[W_i] - (1-\sigma_{i-1})\}$$

$$= \rho_i - a_{i+1} r_i - \{(\overline{W} - \sum_{k=i+1}^{n} \rho_k E[W_k]/E[W_i] - (1-\sigma_i)\}$$

$$= \rho_i - a_{i+1} r_i - b_i .$$

We have thus completely proved (a). We will next show (b). That is, we will show that there is only one positive root of $a_i X^2 + b_i X + c_i = 0$.

As we stated above, this is true if we can show that $a_i\ c_i < 0$. We have already seen that $a_n > 0$ and $c_n < 0$, so we will apply the principle of induction to assume that we have found a_k $(i+1 \leq k \leq n)$ to be positive and c_k $(i+1 \leq k \leq n)$ to be negative; therefore r_k $(i \leq k \leq n-1)$ is positive.

From (2.2.3) it is easily seen that a_i is positive. Thus we only need to show that $c_i < 0$.

Instead of directly proving that $c_i < 0$, we will show the following:

(2.2.14) For $i \leq k \leq n-1$, $c_k < 0$ if and only if

$$E[W_{n-1}]a_n c_n < \sum_{k=i}^{n-1} \rho_k E[W_k](b_{k+1} - \rho_k) .$$

We will prove this by induction.

First, from (2.2.5), we have

$$c_{n-1} = \rho_{n-1} - b_n - r_{n-1} a_n .$$

Recalling that r_{n-1} is the positive root of

$$a_n X^2 + b_n X + c_n = 0 ,$$

we must have

$$r_{n-1} = \frac{1}{2a_n} (-b_n + \sqrt{b_n^2 - 4a_n c_n}) .$$

Thus,

$$c_{n-1} = \rho_{n-1} - b_n - a_n \{ \frac{1}{2a_n} (-b_n + \sqrt{b_n^2 - 4a_n c_n}) \}$$

$$= \rho_{n-1} - b_n - \left(\frac{b_n}{2} + \frac{1}{2}\sqrt{b_n^2 - 4a_nc_n}\right)$$

$$= \rho_{n-1} - \frac{b_n}{2} + \sqrt{\left(\frac{b_n}{2}\right)^2 - a_nc_n} \ .$$

Therefore, $c_{n-1} < 0$ if and only if

$$\rho_{n-1} - \frac{b_n}{2} < \sqrt{\left(\frac{b_n}{2}\right)^2 - a_nc_n}$$

which is equivalent to

$$\rho_{n-1}^2 - \rho_{n-1}b_n + \frac{b_n^2}{4} < \frac{b_n^2}{4} - a_nc_n \ ,$$

or,

$$a_nc_n < \rho_{n-1}(b_n - \rho_{n-1}) \ .$$

This proves (2.2.14) for the case $k = n-1$.

Now, we assume that we have proved (2.2.14) for $k=i+1, i+2, \ldots, n-1$. We will prove (2.2.14) for $k=i$.

Let us consider the quantity

$$E[W_{i-1}]a_ic_i \ .$$

From (2.2.3) and (2.2.5), we have

$$E[W_{i-1}]a_ic_i = E[W_{i-1}]\{(r_ia_{i+1}+\rho_i)E[W_i]/E[W_{i-1}]\}$$

$$\times (\rho_i - b_{i+1} - r_ia_{i+1})$$

$$= E[W_i]\{-(a_{i+1}r_i)^2 - b_{i+1}a_{i+1}r_i + \rho_i(\rho_i - b_{i+1})\}$$

$$= E[W_i]\{-[a_{i+1}(\frac{1}{2a_{i+1}}(-b_{i+1}+\sqrt{b_{i+1}^2-4a_{i+1}c_{i+1}}))$$

$$+\frac{b_{i+1}}{2}]^2+(\frac{b_{i+1}}{2})^2-\rho_i(b_{i+1}-\rho_i)\}$$

$$= E[W_i]\{a_{i+1}c_{i+1}-\rho_i(b_{i+1}-\rho_i)\}$$

$$= E[W_i]a_{i+1}c_{i+1}-\rho_i E[W_i](b_{i+1}-\rho_i) .$$

This is a recursive relation of $E[W_k]a_k c_k$. Therefore, we can recursively substitute the quantity in the right hand side of the above equation to obtain

$$E[W_{i-1}]a_i c_i = E[W_{n-1}]a_n c_n - \sum_{k=i}^{n-1} \rho_k E[W_k](b_{k+1}-\rho_k) .$$

For the quantity $E[W_{i-1}]a_i c_i$, we have already seen that $a_i > 0$. Thus, this quantity is negative if and only if $c_i < 0$. The equation above therefore shows that $c_i < 0$ if and only if

$$(2.2.15) \qquad E[W_{n-1}]a_n c_n - \sum_{k=i}^{n-1} \rho_k E[W_k](b_{k+1}-\rho_k) < 0 .$$

This proves that (2.2.14) holds.

We will now show that (2.2.15) indeed holds. This will then prove that $c_i < 0$, which will complete our proof of (b).

The first term on the left hand side of (2.2.15) is

$$(2.2.16) \qquad E[W_{n-1}]a_n c_n = E[W_{n-1}](\rho_n E[W_n]/E[W_{n-1}])$$

$$\times \{(1-\sigma_{n-1}) - \overline{W}/E[W_n]\}$$

$$= \rho_n E[W_n](1-\sigma_{n-1})-\rho_n\overline{W} .$$

The second term of (2.2.15) is

$$(2.2.17)\qquad \sum_{k=i}^{n-1} \rho_k E[W_k](b_{k+1}-\rho_k)$$

$$= \sum_{k=i}^{n-1} \rho_k E[W_k]\{\overline{W} - \sum_{j=k+1}^{n} \rho_j E[W_j])/E[W_k]-(1-\sigma_k)-\rho_k\}$$

$$= \sum_{k=i}^{n-1} \{\rho_k(\overline{W}- \sum_{j=k+1}^{n} \rho_j E[W_j])-\rho_k E[W_k)(1-\sigma_{k-1})\}$$

$$= (\sum_{k=i}^{n-1} \rho_k)\overline{W} - \sum_{k=i}^{n-1} \sum_{j=k+1}^{n} \rho_k \rho_j E[W_j] - \sum_{k=i}^{n-1} \rho_k E[W_k]$$

$$+ \sum_{k=i}^{n-1} \rho_k \sigma_{k-1} E[W_k]$$

$$= (\sigma_{n-1}-\sigma_{i-1})\overline{W} - \sum_{k=i}^{n-1} \rho_k E[W_k] - \sum_{j=i}^{n} \sum_{k=i}^{j-1} \rho_k \rho_j E[W_j]$$

$$+ \sum_{k=i}^{n-1} \rho_k \sigma_{k-1} E[W_k]$$

$$= (\sigma_{n-1}-\sigma_{i-1})\overline{W} - \sum_{k=i}^{n-1} \rho_k E[W_k] - \sum_{k=i}^{n} \sum_{j=i}^{k-1} \rho_j \rho_k E[W_k]$$

$$+ \sum_{k=i}^{n-1} \rho_k \sigma_{k-1} E[W_k]$$

$$= (\sigma_{n-1}-\sigma_{i-1})\overline{W} - \sum_{k=i}^{n-1} \rho_k E[W_k] - \sum_{k=i}^{n} (\sigma_{k-1}-\sigma_{i-1})\rho_k E[W_k]$$

$$+ \sum_{k=i}^{n-1} \rho_k \sigma_{k-1} E[W_k]$$

$$= (\sigma_{n-1}-\sigma_{i-1}) - \sum_{k=i}^{n-1} \rho_k E[W_k] - (\sigma_{n-1}-\sigma_{i-1})\rho_n E[W_n]$$

$$- \sum_{k=i}^{n-1} (\sigma_{k-1}-\sigma_{i-1})\rho_k E[W_k] + \sum_{k=i}^{n-1} \rho_k \sigma_{k-1} E[W_k]$$

$$= (\sigma_{n-1}-\sigma_{i-1})(\overline{W}-\rho_n E[W_n]) - \sum_{k=i}^{n-1} \rho_k E[W_k] - \sum_{k=i}^{n-1} \sigma_{k-1}\rho_k E[W_k]$$

$$+ \sigma_{i-1} \sum_{k=i}^{n-1} \rho_k E[W_k] + \sum_{k=i}^{n-1} \rho_k \sigma_{k-1} E[W_k]$$

$$= (\sigma_{n-1}-\sigma_{i-1})(\overline{W}-\rho_n E[W_n]) - (1-\sigma_{i-1})(\sum_{k=i}^{n-1} \rho_k E[W_k]).$$

Therefore, (2.2.15) becomes

$$\rho_n E[W_n](1-\sigma_{n-1})-\rho_n\overline{W}-(\sigma_{n-1}-\sigma_{i-1})(\overline{W}-\rho_n E[W_n])$$

$$+ (1-\sigma_{i-1})(\sum_{k=i}^{n-1} \rho_k E[W_k])$$

$$= \rho_n E[W_n]\{(1-\sigma_{n-1})+(\sigma_{n-1}-\sigma_{i-1})\}-\overline{W}\{\rho_n+(\sigma_{n-1}-\sigma_{i-1})\}$$

$$+ (1-\sigma_{i-1})(\sum_{k=i}^{n-1} \rho_k E[W_k])$$

$$= (1-\sigma_{i-1})\rho_n E[W_n]-\overline{W}(\rho-\sigma_{i-1})+(1-\sigma_{i-1}) \sum_{k=i}^{n-1} \rho_k E[W_k]$$

$$= (1-\sigma_{i-1}) \sum_{k=i}^{n} \rho_k E[W_k]-(1-\sigma_{i-1}) \sum_{k=1}^{i-1} \rho_k E[W_k] .$$

Now, from (2.1.4), when $I = \{1,2,\ldots,i-1\}$, we have

$$\sum_{k=1}^{i-1} \rho_k E[W_k] > (1-\rho)\sigma_{i-1}\overline{W}/(1-\sigma_{i-1}) .$$

Therefore

$$(1-\rho)\sigma_{i-1}\overline{W} < (1-\sigma_{i-1}) \sum_{k=1}^{i-1} \rho_k E[W_k] ,$$

or,

$$(1-\rho)\sigma_{i-1}\overline{W} - (1-\sigma_{i-1}) \sum_{k=1}^{i-1} \rho_k E[W_k] < 0 .$$

This shows that (2.2.15) holds, and thus completes the proof of (b).

From Theorem 2.3, we have the following: To determine the values of control parameters $\{\alpha_i\}$ for a given set of $(E[W_1], E[W_2], \ldots, E[W_n])$ satisfying (1.3.2) and (2.2.1), we first rename the class indices so that (2.2.2) is satisfied, and then use the algorithm to obtain the values of the control parameters $\{\alpha_i\}$. Thus, any point $W = (E[W_1],E[W_2],\ldots,E[W_n])$ lying in the space defined by (1.3.2) and (2.2.1) is achievable by escalating priority scheduling. This shows that the feasible performance space of mean waiting times of escalating priority scheduling is the space defined by (1.3.2) and (2.2.1).

The following theorem shows that escalating priority scheduling covers the spectrum from FCFS scheduling to fixed priority scheduling.

THEOREM 2.5. (a) FCFS scheduling can be achieved from escalating priority scheduling by setting the parameters $\{\alpha_i\}$ all equal.

(b) Fixed priority scheduling is the limiting rule of escalating priority scheduling when the ratios between successive parameters all approach infinity.

<u>Proof</u>: (a) When we set all the parameters $\{\alpha_i\}$ equal, we are in fact ignoring the differences among priority classes because all jobs entering the queue have the same priority increasing rate and therefore the next job selected for

service is always the one that has been waiting the longest. Thus, selecting the job with the highest priority index is equivalent to selecting the job that entered the queue the earliest. This is precisely first-come-first-serve scheduling.

(b) Given that $\alpha_i/\alpha_{i+1} \to \infty$ for all i, $1 \leq i < n$, then it is impossible to have any job from class i+1 with priority index larger than any job in the queue from class i. This is because as soon as a job from class i comes into the system, its priority index will increase at such a fast rate that it surpasses all the priority indices of jobs from class i+1, no matter how long they have waited in the queue. Within the same priority class, the job which entered the earliest will have the highest priority index. Thus, first-come-first-serve prevails. Therefore, in this limiting case, we have fixed priority scheduling.

To check this, we will show that

$$(2.1.5) \qquad \lim_{c} E[Wi] = (1-\rho)\,\overline{W} \,/\, (1-\sigma_{i-1})\,(1-\sigma_i),$$

where c is the condition that $\alpha_i/\alpha_{i+1} \to \infty$, $1 \leq i < n$. We will prove this by induction on k (from n backwards to 1).

First, from (1.3.10),

$$\lim_{c} E[Wn] = \overline{W} \,/\, [1 - \sum_{k=1}^{n-1} (1-0)\,\rho_k] = \overline{W} \,/\, (1-\rho+\rho_n) = E[Wn] \ .$$

Next, we assume that (2.1.1) is true for all k = i+1, i+2, ..., n. We show that it is also true for k=i.

Now, for each k>i, it can be seen that

$$\frac{1}{(1-\sigma_{k-1})(1-\sigma_k)} = \frac{1}{\rho_k}(\frac{1}{1-\sigma_k} - \frac{1}{1-\sigma_{k-1}}) .$$

Thus, for $k < i$,

$$\lim_{c} E[W_k] = (1-\rho)\overline{W}/(1-\sigma_{k-1})(1-\sigma_k)$$

$$= \frac{(1-\rho)\overline{W}}{\rho_k}(\frac{1}{1-\sigma_k} - \frac{1}{1-\sigma_{k-1}}) .$$

So, we have

$$\lim_{c} E[W_i] = \frac{\overline{W} - \sum\limits_{k=i+1}^{n} \rho_k \cdot \lim\limits_{c} E[W_k]}{1 - \sum\limits_{k=1}^{i-1} \rho_k}$$

$$= \frac{1}{1-\sigma_{i-1}} \{\overline{W} - \sum_{k=i+1}^{n} (1-\rho)\overline{W} \cdot (\frac{1}{1-\sigma_k} - \frac{1}{1-\sigma_{k-1}})\}$$

$$= \frac{1}{1-\sigma_{i-1}} \cdot \{\overline{W} - \frac{1-\rho}{1-\sigma_n}\overline{W} + \frac{(1-\rho)\overline{W}}{1-\sigma_i}\}$$

$$= \frac{(1-\rho)\overline{W}}{(1-\sigma_{i-1})(1-\sigma_i)} .$$

The above theorem states that the class of escalating priority scheduling covers the spectrum from first-come-first-serve scheduling with no discriminatin to fixed priority scheduling with FCFS rule to determine the order of service for jobs within the same class.

2.3 Interpretations of Boundary Points

In Section 2.2, we showed that the feasible performance space of mean waiting times for escalating priority scheduling is characterized by (1.3.2) and (2.2.1). This is an open set consisting of the interior points of the space defined by (1.3.2) and (2.1.4). In this section, we study the interpretation of the boundary points of the space defined by (1.3.2) and (2.2.1).

In section 2.1, it was shown that the extreme points on the boundary represent mean waiting times behaviors of different orderings of priority classes under fixed priority scheduling. By Theorem 2.5 (b), these points correspond to limiting rules of escalating priority scheduling when $\alpha_i/\alpha_{i+1} \to \infty$ for all i.

We will now investigate the meanings of non-extreme boundary points for a 3-class queueing model using a three dimensional geometrical representation.

In figure 2.3, the points Q_{ijk}'s are the points corresponding to mean waiting times performance of fixed priority scheduling with highest priority given to class i, the next highest to class j, and lowest to class k. The point P is $(\overline{W}, \overline{W}, \overline{W})$ and corresponds to FCFS scheduling. The interior points of the hexagon $Q_{123}Q_{213}Q_{231}Q_{321}Q_{312}Q_{132}$ **are** the open space defined by (1.3.2) and (2.2.1). They are the feasible performance space of mean waiting times under escalating priority scheduling.

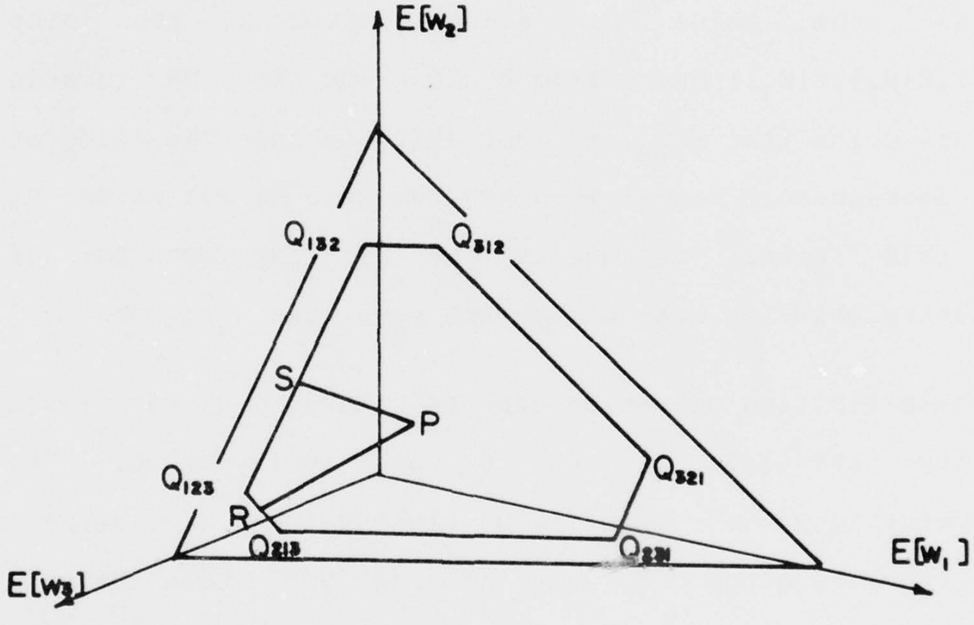


Figure 2.3 Boundary Points of Performance Space in 3-Class Escalating Priority Scheduling

Without loss of generality, we consider the performance space of mean waiting times for the priority ordering that gives class 1 the highest priority, class 2 next, and class 3 the lowest. From Theorem 2.3, this space is the area in the hexagon such that $E[W_1] \leq E[W_2] \leq E[W_3]$. Thus, it is the area bounded by P,R,Q_{123},S in Figure 2.5, where the line PR is defined as $E[W_1]=E[W_2]$ and PS is defined as $E[W_2]=E[W_3]$.

Let us first consider the parametric change of parameters in escalating priority scheduling with $(\alpha_1,\alpha_2,\alpha_3) = (a,a,ak)$ for $k:1\to 0$ (a is any positive constant). Since $\alpha_1=\alpha_2$, we must have $E[W_1]=E[W_2]$. We note that initially, when k=1, we are at the point $P=(\overline{W}, \overline{W}, \overline{W})$. Now, when we decrease the value k, starting from 1, the point $(E[W_1],E[W_2],E[W_3])$ moves from P along the line PR towards R. This means that $E[W_3]$ is increasing because the value of α_3 is decreasing. Hence, when k=0, we must be at point R, and this point represents the limiting behavior of escalating priority scheduling with $\alpha_1=\alpha_2$ and $\alpha_2/\alpha_3 \to\infty$.

This limiting situation can be thought of as fixed priority scheduling applied to two priority groups: The high priority group consisting of classes 1 and 2; the low priority class group consisting of class 3. Within the high priority group, no discrimination is applied between class 1 and class 2 jobs.

Next, consider the parametric setting $(\alpha_1,\alpha_2,\alpha_3) = (a(k+b(1-k)),a,ak)$, where $a>0$ and $b\geq 1$. Again, when we start with k=1, we are at point P. Now, as k decreases, both the relative discriminations between class 1 and class 2, and between class 2 and class 3 increase. This is because α_1/α_2

$= k+b(1-k) \geq 1$ and $\alpha_2/\alpha_3 = 1/k > 1$. This means that the point is moving from P away from PR and toward the boundary RQ_{123} because $E[W_2]$ and $E[W_3]$ are both increasing. Therefore, when k approaches 0, the point must lie on the boundary RQ_{123} and represents the limiting setting of $\alpha_1/\alpha_2=b$ and $\alpha_2/\alpha_3 \to \infty$.

This limiting situation can again be thought of as fixed priority scheduling applied to two groups: the high priority group consisting of classes 1 and 2 and the low priority group consisting of class 3 only. Now however, within the high priority group, class 1 and class 2 jobs compete for service using escalating priority scheduling.

We will now investigate changing the value of b. When b=1 and $k \to 0$, $\alpha_1/\alpha_2=1$ and $\alpha_2/\alpha_3 \to \infty$. We know from our first example that this parameter setting corresponds to the boundary point R. When $b \to \infty$ and $k \to 0$, $\alpha_1/\alpha_2 \to \infty$ and $\alpha_2/\alpha_3 \to \infty$. From Theorem 2.5 (b) we know that this corresponds to the extreme point Q_{123}. Therefore, when b varies between 1 and $< \infty$, $1 \leq \alpha_1/\alpha_2 < \infty$ and $\alpha_2/\alpha_3 \to \infty$ and thus the corresponding limiting point lies on the line segment RQ_{123}.

Similarly, the line segment SQ_{123} represents the limiting points when $\alpha_1/\alpha_2 \to \infty$ and $\alpha_2/\alpha_3=b$ for $1 \leq b < \infty$. This can be seen using the parametric setting $(\alpha_1, \alpha_2, \alpha_3) = (a/k, a, a(k+(1-k)/b)$ and using the arguments presented above for line segment RQ_{123}.

Therefore, non-extreme points on the boundary can be thought of as representing the mean waiting times performance of priority scheduling rules with a "mixture" of fixed priority scheduling and escalating priority scheduling. We now formally define this new rule of scheduling:

DEFINITION. In multiclass queueing systems, a scheduling rule is called mixed priority scheduling of escalating priority and fixed priority, or, MEFP scheduling in short, if it separates the arrival classes into several priority levels such that fixed priority scheduling is applied between different levels, and escalating priority scheduling is applied within levels.

This scheduling mechanism is shown in Figure 2.4. Operationally, let the n arriving classes be grouped into m levels, eacn level i consisting of n_i priority classes. For each priority class (i,j), there is a control parameter α_{ij} associated with it. Jobs within the level i compete for service with priorities increasing with rate α_{ij}. When the server becomes free, it searches from the top level downwards to find the first level with at least one nonempty queue. It then select the job within this level with the highest priority index for service.

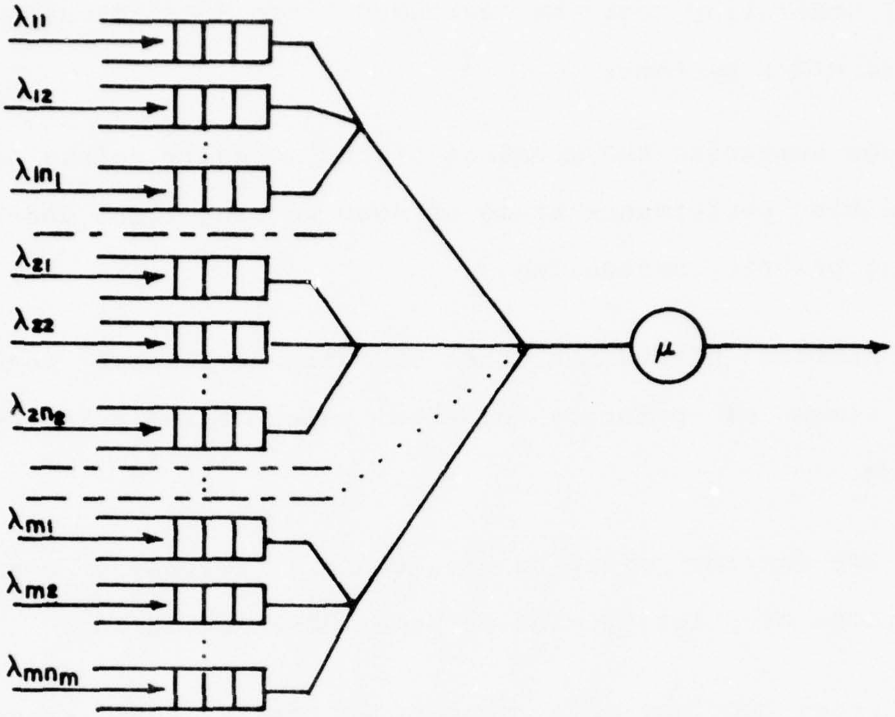


Figure 2.4 MEFP Scheduling Mechanism

Notice that in this scheduling rule, within each level i, we have n_i-1 degrees of freedom for controlling the parameters $\{\alpha_{ij}\}$. Thus, the total degrees of freedom for controlling this scheduling is n-m.

Kleinrock first studied this limiting situation of escalating priority scheduling in [KLEI66] and called it Strict and Lag Priority Mixture (SLPM). He derived mean waiting times behavior for M/M/1 queueing systems under SLPM scheduling. We note that his results for MEFP (SLPM) scheduling are also true for M/G/1 queueing systems for the reasons given in Section 1.3 that his results of escalating priority scheduling can be extended from M/M/1 queueing systems to M/G/1 systems.

We now summarize the meanings of the boundary points of the feasible performance space of mean waiting times under escalating priority scheduling:

(a) Extreme points on the boundary represent mean waiting times of priority classes under fixed priority scneduling.

(b) Non-extreme points on the boundary represent mean waiting times of priority classes under MEFP scheduling.

Before we conclude this chapter, we remark that since boundary points are limiting behaviors of escalating priority scheduling, we can modify Theorem 2.4 to obtain a set of values for escalating priority scheduling that can approximate the performance of fixed priority and MEFP scheduling. This is as follows:

Given a set of mean waiting times performance of either fixed priority or MEFP scheduling, they correspond to a boundary point of the space defined by (1.3.2) and (2.1.4). When we solve the quadratic equation $a_iX^2 + b_iX + c_i = 0$ following Theorem 2.4, instead of having a unique positive root, we will have a unique non-negative root. In this case, (2.2.10) can not be directly applied to determine the values of $\{ \alpha_i \}$ because the denominator of the expression in

(2.2.10) is zero. However, if we recall from (2.2.11) that r_i is defined as the ratio α_{i+1}/α_i, then $r_i=0$ means that the value α_i/α_{i+1} must approach infinity. Thus, in order to obtain a set of parameters such that $\alpha_i/\alpha_{i+1}\rightarrow\infty$, we can replace $r_i=0$ by $r_i=\epsilon$ where ϵ is a very small positive quantity so that $\alpha_i/\alpha_{i+1}=1/\epsilon$ will be large. Using this substitution, we can then obtain a set of positive r_i's and therefore (2.2.10) of Theorem 2.4 can be used to obtain a set of values $\{\alpha_i\}$. Therefore, using this set of values for the control parameters in escalating priority scheduling, fixed priority and MEFP schedulings can be approximated.

CHAPTER 3

OPTIMIZATION OF ESCALATING PRIORITY SCHEDULING

3.0 Introduction

In Chapter 2, we showed that escalating priority scheduling can cover the spectrum of scheduling from FCFS scheduling to fixed priority scheduling by adjusting the control parameters $\{\alpha_i\}$ and that the feasible performance space of the $E[W_i]$'s is defined by (1.3.2) and (2.2.1). In this chapter, we present algorithms for determining the values of $\{E[W_i]\}$ to optimize various cost functions. We then can use the algorithm presented in Theorem 2.4 to obtain the values of the control parameters $\{\alpha_i\}$ in escalating priority scheduling to give the values of mean waiting times determined from these optimizing algorithms.

We will develop optimization algorithms for two general cost objective functions. Each of these general functions has several special cases of interest. Two new variables, s_i and c_i, are used in these general functions. s_i is a constant for class i and its meaning differs for each of the various cases. c_i represents the appropriate cost of each job in class i for the objective function being used. The two general objective functions are the following:

$$\text{(a) Minimize } \sum_{k=1}^{n} \lambda_k c_k (E[W_k]-s_k)^m, \qquad (m\geq 1)$$

and (b) Minimize max $\{\lambda_k c_k(E[W_k]-s_k);\ 1\leq k\leq n\}$.

We will now present some of the interesting special cases of the general objective functions.

Case 1. For objective function (a) with m=1 and $s_i=-E[S_i]$, we have:

$$\text{Minimize } \sum_{k=1}^{n} \lambda_k c_k(E[W_k]+E[S_k]),$$

where c_i is the cost of the time spent in the system per unit time for each job from class i. This objective function minimizes the expected total cost of the average time jobs spend in the system (waiting plus service time).

Case 2. For the objective function (a) with m=1 and $s_i=0$ for all i, we have:

$$\text{Minimize } \sum_{k=1}^{n} \lambda_k c_k E[W_k],$$

where c_i is the cost of waiting per unit time for each job from class i. This objective function minimizes the expected total cost of the average time jobs spend waiting for service.

Case 3. For the objective function (a) with m=2 and $s_i=-E[S_i]$, we have:

$$\text{Minimize } \sum_{k=1}^{n} \lambda_k c_k(E[W_k]+E[S_k])^2,$$

where c_i is the cost of the time spend in the system per unit time squared for each job from class i. This objective function minimizes the expected total cost of the square of the average time jobs spend in the system.

Case 4. For the objective function (a) with m=2 and $s_i=0$ for all i, we have:

$$\text{Minimize} \sum_{k=1}^{n} \lambda_k c_k E[W_k]^2,$$

where c_i is the cost of waiting per unit time squared for each job from class i. This objective function minimizes the expected total cost of the square of the average time jobs spend waiting.

Case 5. For the objective function (b), if we interpret s_i as the slack time allowed for jobs from class i, then $E[W_i]-s_i$ is the expected lateness of a job, and c_i is the cost per unit time for each job late in class i. Thus this objective function minimizes the expected lateness cost of the average lateness of all jobs in the class having the maximum expected lateness cost.

From optimization theory, optimal solution(s) exist for any closed and bounded feasible solution domain. Since the feasible performance space of mean waiting times of escalating priority scheduling is an open space defined by (1.3.2) and (2.2.1), we will add the boundary points of this space to the solution domain to guarantee that optimal solution(s) exist. Thus, for our optimization problems, the constraints are (1.3.2) and (2.1.4). For these problems, if the optimal solution(s) obtained falls on a boundary, we can use the modification of Theorem 2.4 described in Section 2.3 to obtain a set of $\{\alpha_i\}$ that will approximate the optimal mean waiting times.

In the following sections, Section 3.1 discusses optimization of (a) with m=1 and (b) and Section 3.2 discusses the optimization of (a) with m>1. In Section 3.3, we present a series of examples to illustrate the optimization techniques described in Sections 3.1 and 3.2.

3.1 Optimization with Linear Cost Functions

In this section, we discuss optimizing the following objective functions:

(3.1.1) Minimize $\sum_{k=1}^{n} \lambda_k c_k (E[W_k]-s_k)$

and

(3.1.2) Minimize $\max\{\lambda_k c_k(E[W_k]-s_k);\ \ 1\leq k\leq n\}$

subject to

(1.3.2) $\rho_1 E[W_1] + \rho_2 E[W_2] + \ldots + \rho_n E[W_n] = \rho\overline{W},$

and

(2.1.4) $\forall I \subset N \quad \sum_{k\in I} \rho_k E[W_k] \geq (1-\rho)(\sum_{k\in I} \rho_k)\overline{W}/(1-\sum_{k\in I}\rho_k).$

We first discuss (3.1.1). For this objective function, we have a bona fide linear programming problem. Thus, an optimal solution can be obtained by the simplex method. From the theory of linear programming, if a single optimal solution exists, it must occur at an extreme point. When multiple optimal solutions exist, they must be convex combinations of optimal extreme points.

A solution to this optimization problem can be obtained directly without using the simplex method because we know that extreme points represent fixed priority scheduling. We recall from Theorem 1.4 that for fixed priority scheduling, to minimize

$$(1.3.7) \qquad \sum_{k=1}^{n} \lambda_k c_k E[W_k],$$

the optimal ordering of priority classes follows the descending order of $\mu_i c_i$. If for some i and j, $\mu_i c_i = \mu_j c_j$, then multiple optimal solutions exist and the ordering of these two classes is unimportant.

Now, since the objective function (3.1.1) can be rewritten as

$$\sum_{k=1}^{n} \lambda_k c_k (E[W_k]-s_k) = \sum_{k=1}^{n} \lambda_k c_k E[W_k] - \sum_{k=1}^{n} \lambda_k c_k s_k,$$

optimizing (3.1.1) is equivalent to optimizing (1.3.7) when the two problems have the same solution domain. This is because the last summation term is a constant independent of the E[Wi]'s.

Even though the solution space for optimizing (3.1.1) contains points other than the extreme points that correspond to fixed priority scheduling, we can temporarily restrict ourselve to this set of extreme points to obtain initial optimal solution(s). Under this condition, we can use Theorem 1.4 to obtain initial optimal solution(s) and then take convex combinations of all the initial optimal solution(s) to obtain multiple optimal solutions. A summary of the above discussion is the following:

To optimize (3.1.1), first order the $\mu_i c_i$'s in descending sequence.

Case (a): If no two $\mu_i c_i$'s are equal, then the optimal solution to (3.1.1) is unique. Denoting the class indices for the descending sequence of $\mu_i c_i$ as 1', 2', ..., n', then from Theorem 1.3, we have

$$(3.1.3) \qquad E[W_{i'}]^* = (1-\rho)\ \overline{W}\ /\ (1-\sigma_{i-1'})\ (1-\sigma_{i'}),$$

Case (b): If some of the $\mu_i c_i$'s are equal, then for all non-ascending orderings of $\mu_i c_i$, we first use (3.1.3) to obtain all the optimal solutions within the domain of fixed priority scheduling. Then, the solutions to (3.1.1) are the set of all convex combinations of the initial optimal solutions.

Next, we discuss optimization of (3.1.2) subject to (1.3.2) and (2.1.4).

To solve this, we employ the following argument: Given an objective function f(x) with solution domain X, instead of **searching directly** through the whole domain of X for the minimum, we decompose X into several subregions and first determine the minimum within each subregion. Then, the optimal solution to the original problem is the solution of the subspace that has the smallest value for the objective function among the minimum values found for all the subspaces. Mathematically, we have:

$$(3.1.4) \qquad \min_{x\in X} f(x) = \min_{i\in I} \min_{x\in X_i} f(x) \qquad \text{if } X = \bigcup_{i\in I} X_i.$$

Now, for our optimization problem (3.1.2), we decompose our solution space R defined by (1.3.2) and (2.1.4) into n subregions $R_1, R_2, \ldots, R_n$ such that each region R_i is defined by (1.3.2), (2.1.4) and the following set of n-1 constraints:

$$(3.1.5) \qquad \lambda_i c_i(E[W_i]-s_i) \geq \lambda_k c_k(E[W_k]-s_k) \qquad \text{for all } k\neq i.$$

We note that for each i, (3.1.5) is a different set of linear inequalities and it means that for each point W = $(E[W_1],E[W_2],\ldots,E[W_n])$ in R_i, if we consider the values $\lambda_1 c_1(E[W_1]-s_1)$, $\lambda_2 c_2 (E[W_2]-s_2)$, .., $\lambda_n c_n(E[W_n]-s_n)$, then $\lambda_i c_i(E[W_i]-s_i)$ is the largest among them. That is, for each $W\in R_i$,

$$\max \{\lambda_k c_k(E[W_k]-s_k); 1\leq k\leq n\} = \lambda_i c_i(E[W_i]-s_i).$$

Now, since for each point W in R, if we consider the values $\lambda_1 c_1(E[W_1]-s_1)$, $\lambda_2 c_2(E[W_2]-s_2),\ldots,$ $\lambda_n c_n(E[W_n]-s_n)$, then one of them must be the largest. If, say, $\lambda_j c_j(E[W_j]-s_j)$ is the largest among the above, this means that the point W must lie in region R_j. Thus, each point in R must lie in one of the regions R_i. This proves that

$$R_1 \cup R_2 \cup \ldots \cup R_n = R.$$

We now define $f(W) = \max \{\lambda_k c_k (E[W_k]-s_k); \quad 1 \leq k \leq n\}$. Then our minimization problem (3.1.2) becomes $\min\limits_{W \in R} f(W)$. From (3.1.4), this problem is then equivalent to $\min\limits_{1\leq i\leq n} \min\limits_{W\in R_i}$ f(W), which is

$$(3.1.6) \qquad \min_{1\leq i\leq n} \min_{W\in R_i} \max \{\lambda_k c_k (E[W_k]-s_k); \quad 1 \leq k \leq n\}.$$

Since for each point W within region R_i, we have

$$\max \{\lambda_k c_k (E[W_k]-s_k); 1 \leq k \leq n\} = \lambda_i c_i (E[W_i]-s_i),$$

substituting this into (3.1.6), we obtain

$$\min_{1\leq i\leq n} \min_{R_i} \lambda_i c_i (E[W_i]-s_i).$$

The above shows that we can solve (3.1.2) by the following two-step procedure:

Step (1): For each i, $1 \leq i \leq n$, solve the following:

$$(3.1.7) \qquad \text{Minimize } \lambda_i c_i (E[W_i]-s_i)$$

subject to (1.3.2), (2.1.4) and (3.1.5).

Step (2): Select, among all the optimal solutions obtained from step (1), the one with the minimum objective value. This solution is then the optimal solution to the original problem (3.1.2).

We remark that in the step (1) above, n linear programming problems need to be solved, each one using a subregion of the original solution space defined by the constraint sets (1.3.2), (2.1.4) and (3.1.5). For each different i, (3.1.5) defines a different region R_i. It is possible that under the partition used, some subregions may be empty so that feasible solutions for these problems do not exist. However, since at least one of the regions R_i is always nonempty, a solution to our original problem exist.

Before we conclude this section, we point out the following interesting observation: For the optimization of (3.1.2), if the optimal solution falls in the interior of the solution domain, then the following relation must hold:

$$\lambda_1 c_1(E[W_1]-s_1)=\lambda_2 c_2(E[W_2]-s_2)=\ldots=\lambda_n c_n(E[W_n]-s_n). \tag{3.1.8}$$

This is because if $W^* = (E[W_1]^*, E[W_2]^*, \ldots, E[W_n]^*)$ were an optimal solution but did not satisfy (3.1.8), then if we rename the class indices as 1', 2', ..., n' such that

$$\begin{aligned}\lambda_{1'}c_{1'}(E[W_{1'}]-s_{1'}) &\geq \lambda_{2'}c_{2'}(E[W_{2'}]-s_{2'})\\ &\geq \ldots \geq \lambda_{n'}c_{n'}(E[W_{n'}]-s_{n'}),\end{aligned} \tag{3.1.9}$$

then at least one of the above "$\geq$" must have strictly greater relation ">". Let class i' and i+1' be the smallest indices such that

$$\lambda_{i'} c_{i'} (E[W_{i'}]-s_{i'}) > \lambda_{i+1'} c_{i+1'} (E[W_{i+1'}]-s_{i+1'}).$$

That is, we have

$$\lambda_{1'} c_{1'} (E[W_{1'}]-s_{1'}) = \lambda_{2'} c_{2'} (E[W_{2'}]-s_{2'}) = \ldots = \lambda_{i'} c_{i'} (E[W_{i'}]-s_{i'}).$$

Then, from the conservation law of mean waiting times, it is possible to further reduce the $E[W_{k'}]$'s, $1 \leq k \leq i$, by increasing $E[W_{k'}]$, $i+1 \leq k \leq n$ so that $\lambda_{1'} c_{1'} (E[W_{1'}]-s_{1'})$ will be decreased. This contradicts our assumption that W* is optimal.

We note that the above argument does not apply to the boundary points because if we try to reduce the values of some of the $E[W_{i'}]$'s, then they may be forced off the boundary and become infeasible.

The above shows that if we temporarily neglect the boundary constraints (2.1.4), then (3.1.8) can be used to obtain a solution to the relaxed problem (3.1.2) with only the constraint (1.3.2). Thus, for our original problem, instead of following the aforementioned procedure of solving n linear programming problems, we can first solve (3.1.8) and (1.3.2) to obtain a set of $\{E[W_i]\}$, and then test to see if the constraint set (2.1.4) is satisfied. If the solution does satisfy the constraint set (2.1.4), then it is an optimal solution to our original problem. However, if some of the constraints in (2.1.4) are violated, then we must use the original procedure to obtain a solution.

We now return to the problem of solving (3.1.8) and (1.3.2) for $\{E[W_i]\}$. First, let the common value of (3.1.8) be t, so that

$$E[W_i] = s_i + t / \lambda_i c_i.$$

Substitute this into (1.3.2), we have

$$\sum_{k=1}^{n} \rho_k (s_k + t/\lambda_k c_k) = \rho\overline{W},$$

or,

$$t = (\rho \overline{W} - \sum_{k=1}^{n} \rho_k s_k) / (\sum_{k=1}^{n} 1/\mu_k c_k).$$

Thus, we have

$$E[W_i] = s_i + (\rho\overline{W} - \sum_{k=1}^{n} \rho_k s_k)/(\lambda_i c_i \sum_{k=1}^{n} 1/\mu_k c_k). \qquad (3.1.10)$$

This is then the solution to the relaxed problem (3.1.2) with constraint (1.3.2) only. If it also satisfies (2.1.4), then it is the optimal solution to (3.1.2). However, if it fails to satisfy all the inequalities in (2.1.4), then we must use our original solution procedure.

We now illustrate this with a two-class priority queueing example.

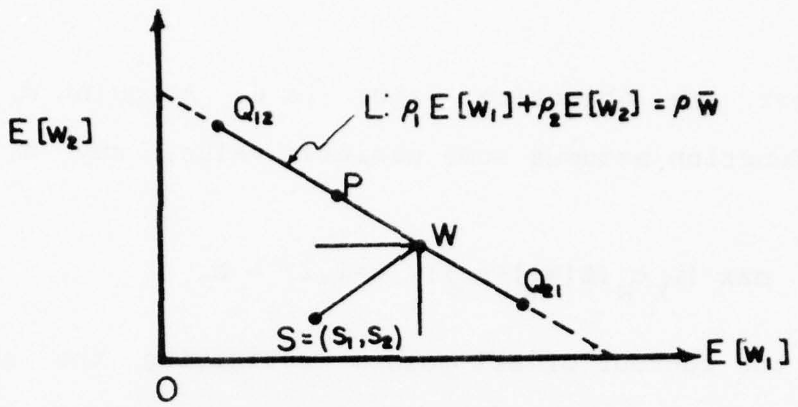


Figure 3.1 Minimize max $\{\lambda_k c_k (E[W_k]-s_k);\ k=1,2\}$

In Figure 3.1, our solution space for optimization is the line segment $\overline{Q_{12}Q_{21}}$. Using (3.1.10) and the given positive values for s_1 and s_2, we have:

$$E[W_1] = s_1 + \frac{\rho\bar{W} - \rho_1 s_1 - \rho_2 s_2}{\lambda_1 c_1 (\frac{1}{\mu_1 c_1} + \frac{1}{\mu_2 c_2})},$$

And,

$$E[W_2] = s_2 + \frac{\rho\bar{W} - \rho_1 s_1 - \rho_2 s_2}{\lambda_2 c_2 (\frac{1}{\mu_1 c_1} + \frac{1}{\mu_2 c_2})}.$$

Thus, if $W = (E[W_1], E[W_2])$ lies in between Q_{12} and Q_{21}, then it is an optimal solution. However, if it does not, then we need to use the two-step procedure that we described earlier.

In this example, the point S is defined as (s_1, s_2). It is interesting to note the following behavior of the objective function: At the point S, $E[W_1] = s_1$ and $E[W_2]$ =

s_2, so that the objective value is 0. At point W, the objective function assumes some positive value, say b, so that

$$\max \{\lambda_k c_k (E[W_k]-s_k); \quad k=1,2\} = b.$$

Thus, the contour of all points satisfying the above equation is the two "half lines" passing through the point W that are parallel to the two axes.

3.2 Optimization with Nonlinear Cost Functions

In this section, we discuss priority scheduling subject to non-linear cost of delay. Our problem is the following:

$$(3.2.1) \qquad \text{Minimize} \quad \sum_{k=1}^{n} \lambda_k c_k \,(E[W_k] - s_k)^m \quad (m>1)$$

subject to

$$(1.3.2) \qquad \rho_1 E[W_1] + \rho_2 E[W_2] + \ldots + \rho_n E[W_n] = \rho\overline{W},$$

and

$$(2.1.4) \qquad \forall I \subset N \quad \sum_{k\in I} \rho_k E[W_k] \geq (1-\rho)(\sum_{k\in I} \rho_k)\overline{W}/(1-\sum_{k\in I} \rho_k).$$

First we discuss the case when m=2. With a quadratic objective function and linear constraints, we have a quadratic programming problem.

This quadratic constrained optimization problem can be solved based on a theorem developed by Kuhn and Tucker [KUHN51], which we will now state in the following. (The proof of this theorem can be found in most nonlinear programming books.)

THEOREM 3.1. Consider the problem

$$\text{maximize } Zo = f\ (Z_1, Z_2, \ldots, Z_n)$$

subject to

(1) $g_i(Z_1, Z_2, \ldots, Z_n) = 0$ for $i = 1,2,\ldots,\ell$,

(2) $h_j(Z_1, Z_2, \ldots, Z_n) \geq 0$ for $j = 1,2,\ldots,m$,

where f, g_i, h_j, are all continuously differentiable functions. Then,

$Z^* = (Z_1, Z_2, \ldots, Z_n)$ can be an optimal solution to the non-linear optimization problem only if there exists $\ell + m$ multipliers $x_1, x_2, \ldots, x_\ell$, and $y_1, y_2, \ldots, y_m$ such that the following conditions are all satisfied:

(3.2.2) $$\frac{\partial f}{\partial Z_k}(Z^*) - \sum_{i=1}^{\ell} x_i \cdot \frac{\partial g_i}{\partial Z_k}(Z^*) - \sum_{j=1}^{m} y_j \cdot \frac{\partial h_j}{\partial Z_k} = 0 \quad (1 \leq k \leq n)$$

(3.2.3) $g_i\ (Z^*) = 0$ for $i = 1, 2, \ldots, \ell$,

(3.2.4) $h_j\ (Z^*) \geq 0$ for $j = 1, 2, \ldots, m$,

(3.2.5) $y_j\ h_j\ (Z^*) = 0$ for $j = 1, 2, \ldots, m$,

(3.2.6) $y_j \leq 0$ for $j = 1, 2, \ldots, m$.

Furthermore, these conditions (called Kuhn-Tucker conditions) are also sufficient if the following are satisfied:

(3.2.7) $f(Z)$ is concave,

(3.2.8) $g_i(Z)$ is linear, for all $i = 1, 2, \ldots, \ell$,

(3.2.9) $h_j(Z)$ is concave, for all $j = 1, 2, \ldots, m$.

In order to apply this theorem, we use the following matrix notations:

Let $\underline{A} = (Aij)$ be a $(2^n-2) \times n$ matrix defined by

(3.2.10) $$Aij = \begin{cases} \rho_j & \text{if } 0 \leq (i \bmod 2^j) < 2^{j-1}, \\ 0 & \text{if } 2^{j-1} \leq (i \bmod 2^j) < 2^j. \end{cases}$$

where the notation "x mod y" is used to represent the remainder of x divided by y.

Let $\underline{b} = (b_1, b_2, \ldots, b_{2^n-2})^T$ be defined by

(3.2.11) $$bi = (1 - \rho)\, \overline{W} \left(\sum_{k=1}^{n} A_{ik} \right) / \left(1 - \sum_{k=1}^{n} A_{ik}\right).$$

also, denote $\underline{W} = (E[W_1], E[W_2], \ldots, E[Wn])$, and

$$\underline{\rho} = (\rho_1, \rho_2, \ldots, \rho_n) .$$

Then, if we let

(3.2.12) $g(\underline{W}) = \underline{\rho}^T\, \underline{W} - \rho\, \overline{W}$, and

(3.2.13) $\underline{h}(\underline{W}) = \underline{A}\ \underline{W} - \underline{b}$,

we see that constraints (1.3.2) and (2.1.4) can be written as $g(\underline{W}) = 0$ and $\underline{h}(\underline{W}) \geq 0$ respectively.

Now, in order to change our minimization problem to a maximization problem, we define f(W) to be the negative of our objective function. That is, we let

$$f(\underline{W}) = - \sum_{k=1}^{n} \lambda_k c_k \ (E[W_k] - s_k)^2 .$$

If we denote

$$\underline{s} = (s_1, s_2, \ldots, s_n)^T ,$$

and let $\underline{D}$ = (Dij) be an n x n matrix defined by

$$Dij = \begin{cases} - \lambda_i c_i & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

then, $f(\underline{W})$ can be written as

(3.2.14) $f(\underline{W}) = \underline{s}^T \underline{D}\ \underline{s} - 2\ \underline{s}^T \underline{D}\ \underline{W} + \underline{W}^T \underline{D}\ \underline{W}$.

From the definitions of the functions f, g, h above, it is easily seen that conditions (3.2.7) through (3.2.9) are satisfied. Thus, the Kuhn-Tucker conditions (3.2.2) through (3.2.6) become sufficient in providing optimal solutions.

We will now derive these Kuhn-Tucker conditions for our problem and then see how the solution of this set of relations can be obtained by applying a modified linear programming method using the artificial variable technique.

First, substitute (3.2.12) through (3.2.14) into (3.2.2), we have:

$$- 2\ \underline{s}^T\ \underline{D} + 2\ \underline{W}^T\ \underline{D} - x\underline{\rho}^T - \underline{y}^T\ \underline{A} = 0.$$

Let $x = x_1 - x_2$ be the difference of two non-negative variables and let $\underline{y}' = - \underline{y}$, substituting into the above equation and taking the transpose, keeping in mind that $\underline{D}^T = \underline{D}$, we obtain

$$2\ \underline{D}\ \underline{W} - \underline{\rho}\, x_1 + \underline{\rho}\, x_2 + \underline{A}^T\underline{y}' = 2\ \underline{D}\ \underline{s}.$$

Denote $\underline{x}' = (x_1, x_2)$, then in matrix notation, the above equation becomes

(3.2.15) $$(2\ \underline{D} \mid -\underline{\rho}, \underline{\rho} \mid \underline{A}^T) (\underline{W} \mid \underline{x}' \mid \underline{y}') = 2\ \underline{D}\ \underline{s}.$$

Next, substitute (3.2.12) into (3.2.3), we have

(3.2.16) $$\underline{\rho}^T\ \underline{W} = \rho\,\overline{W}.$$

Then, substitute (3.2.13) into (3.2.4), we obtain

(3.2.17) $$\underline{A}\ \underline{W} \geq \underline{b}.$$

Now, if we let $\underline{T} = \underline{h}(\underline{W}) = \underline{A}\ \underline{W} - \underline{b}$ be non-negative slack variables for the inequalities (3.2.17), then (3.2.5) becomes

$$\underline{y}\ \underline{h}\ (\underline{W}) = y_j T_j = - y'_j\, T_j = 0$$

or, equivalently,

(3.2.18) $\quad y'_j\, T_j = 0, \qquad$ for $j = 1, 2, \ldots, 2^n-2$.

We recall that in our problem of determining optimal $\underline{W} = (E[W_1], E[W_2], \ldots, E[Wn])$, each $E[W_i]$ is required to be positive. Also, $\underline{x}'$ is defined to be non-negative. By the definition of $\underline{y}'$ and (3.2.6), $\underline{y}'$ is also non-negative. Thus, we have

(3.2.19) $\quad \underline{W}, \underline{x}', \underline{y}', \underline{T} \geq 0.$

Therefore, the Kuhn-Tucker conditions for our problem can be summarized as (3.2.15) through (3.2.19).

To solve the simultaneous equations and inequalities (3.2.15) through (3.2.19), we apply the artificial variables technique, and introduce to each linear equality and "greater or equal" constraints a non-negative artificial variable to formulate a pseudo minimization problem with the sum of introduced artificial variables as the objective function. According to this technique, if the minimum value of this pseudo objective function is zero (which indicates that all the introduced artificial variable have vanished), then the original problem is feasible, and an initial optimal solution can be directly obtained from the solution of the pseudo optimization problem (c.f., any standard linear programming book).

For our problem of seeking an optimal solution to (3.2.14) with constraints (3.2.15) through (3.2.19), we will now temporarily neglect (3.2.18) and (3.2.19), and add

$$n + 1 + (2^n - 2) = 2^n + n - 1$$

artificial variables R_1 through R_{2^n+n-1} to constraints (3.2.15), (3.2.16) and (3.2.17) and obtain a minimization problem in the following:

$$\text{Minimize} \quad \sum_{k=1}^{2^n+n-1} R_k$$

subject to (3.2.18), (3.2.19) and

$$(3.2.20) \qquad (2\underline{D} \mid -\underline{\rho}, \underline{\rho} \mid \underline{A}^T \mid \underline{0} \mid \underline{I}, \underline{0}, \underline{0}) \; (\underline{W} \mid \underline{x}' \mid \underline{y}' \mid \underline{T} \mid \underline{R}) \quad = 2\underline{Ds},$$

$$(3.2.21) \qquad (\underline{\rho}^T \mid \; \underline{0}, \underline{0} \mid \; \underline{0} \mid \underline{0} \mid \underline{0}, \underline{I}, \underline{0}) \; (\underline{W} \mid \underline{x}' \mid \underline{y}' \mid \underline{T} \mid \underline{R}) \quad = \rho\overline{W},$$

$$(3.2.22) \qquad (\; \underline{A} \mid \; \underline{0}, \underline{0} \mid \underline{0} \mid -\underline{I} \mid \underline{0}, \underline{0}, \underline{I}) \; (\underline{W} \mid \underline{x}' \mid \underline{y}' \mid \underline{T} \mid \underline{R}) \quad = \underline{b}.$$

Except for the constraint (3.2.18), the constraints are linear. The simplex method can be used with a slight modification to solve this problem. First, temporarily neglect (3.2.18) and formulate a linear programming problem. Then, we use the simplex method in solving this problem with the additional requirement that for every iteration of generating a new basic feasible solution, we check (3.2.18) to be sure that only one of the two variables y_j' and T_j are a basic variable. That is, if one of the two variables are already a basic variable, then the other variable cannot be the variable entering the basis unless the other variable is the variable leaving the basis. This guarantees that the solution obtained satisfies (3.2.18).

We have now completely solved our quadratic optimization problem. It should be noted, however, that the above solution procedure involves solving a programming problem with a large number of variables, in particular, when the number of priority classes is large. An alternative solution method is described below that may be simpler than using the above algorithm. This solution method is particularly useful when the optimal solution lies in the interior of the feasible performance space.

In the following, we discuss the objective function (3.2.1) in general (m not restricted to being equal to two), and describe an iterative method for solving the optimization of (3.2.1). This iterative method can be simply stated as follows:

First, solve the problem of optimization with some of the constraints relaxed (temporarily neglected). After a solution to the relaxed problem is obtained, check to determine if this solution satisfies the neglected constraints. If all the constraints are satisfied, then this is the optimal solution to the original problem. However, if some of the constraints are violated, then the violated constraints are added and the problem is solved again. This process is repeated until a solution satisfying all constraints is found.

To solve our optimization problem (3.2.1), we first relax all the inequality constraint (2.1.4) and only keep the constraint (1.3.2). This relaxed constraint optimization problem can be solved using Lagrange's multiplier method.

Let $$L = \sum_{k=1}^{n} \lambda_k c_k (E[W_k]-s_k)^m - \ell(\sum_{k=1}^{n} \rho_k E[W_k]-\rho\overline{W}) .$$

Lagrange's necessary condition for optimality is:

(3.2.23) $\frac{\partial L}{\partial \ell} = 0$, and

(3.2.24) $\frac{\partial L}{\partial E[W_k]} = 0$, for $1 \le k \le n$.

Now, for each i, i= 1, 2, ..., n,

$$\partial L/\partial E[W_k] = m\lambda_i c_i (E[W_i]-s_i)^{m-1} - \ell\rho_i .$$

Setting them equal to 0, we obtain

(3.2.25) $E[W_i] = s_i + (\ell / m \ \mu_i c_i)^{1/(m-1)}$. $(1 \le i \le n)$

The first Lagrange condition $\partial L/\partial \ell = 0$ requires

(1.3.2) $\sum_{k=1}^{n} \rho_k E[W_k] = \rho\overline{W}$.

Substitute (3.2.25) into the above equation, we get

$$(\sum_{k=1}^{n} \rho_k s_k) + (\frac{\ell}{m})^{\frac{1}{m-1}} (\sum_{k=1}^{n} \rho_k / (\mu_k c_k)^{\frac{1}{m-1}}) = \rho\overline{W} .$$

Thus,

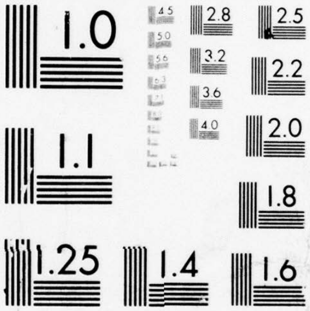
$$(\frac{\ell}{m})^{\frac{1}{m-1}} = (\rho\overline{W} - \sum_{k=1}^{n} \rho_k s_k) / (\sum_{k=1}^{n} \rho_k / (\mu_k c_k)^{\frac{1}{m-1}}) .$$

AD-A068 160 SYRACUSE UNIV NY SCHOOL OF COMPUTER AND INFORMATION --ETC F/G 12/1
A DYNAMIC PRIORITY SCHEDULING RULE WITH OPTIMIZATION FOR TRANSA--ETC(U)
FEB 79 D K WOOD F30602-77-C-0235

UNCLASSIFIED RADC -TR-79-8 NL

2 OF 2
AD
A068160

END
DATE
FILMED
6 --79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

We therefore have, for i = 1, 2, ..., n,

$$(3.2.26) \qquad E[W_i] = s_i + (\tfrac{\ell}{m})^{\frac{1}{m-1}} / (\mu_i c_i)^{\frac{1}{m-1}}$$

$$= s_i + (\rho\overline{W} - \sum_{k=1}^{n} \rho_k s_k)/\{(\mu_i c_i)^{\frac{1}{m-1}} \sum_{k=1}^{n} \rho_k/(\mu_k c_k)^{\frac{1}{m-1}}\} .$$

This solution can now be substituted into (2.1.4) to see if all the constraints are satisfied. If they are, then it is the optimal solution. IF not, we add the constraints that are violated and solve the more restricted mathematical programming problem. In the case when m=2, we have a quadratic programming problem as above with less constraints. If m>2, we have a convex objective function and a set of linear constraint. There are various ways of solving this type of optimization problem. For example, see [TAHA76]. One iteration of adding constraints should normally solve the problem. The resulting solution will be on the boundary and will be either fixed priority or MEFP scheduling.

Now let us consider the special case of (3.1.2) when all $s_i = 0$. Using (3.2.26) as the solution to the relaxed problem (3.1.2) with constraint (1.3.2) only, we have

$$(3.2.27) \qquad E[W_1] : E[W_2] : \dots : E[Wn]$$

$$= (\frac{1}{\mu_1 c_1})^{\frac{1}{m-1}} : (\frac{1}{\mu_2 c_2})^{\frac{1}{m-1}} : \dots : (\frac{1}{\mu_n c_n})^{\frac{1}{m-1}} .$$

Thus, for any i and j,

$$E[W_i]/E[W_j] = (\mu_j c_j/\mu_i c_i)^{\frac{1}{m-1}} .$$

And, we have:

(3.2.28) $\quad E[W_i] < E[W_j] \quad$ if and only if $\mu_i c_i > \mu_j c_j$.

This result shows that for an optimal solution to the problem

(3.2.29) $$\sum_{k=1}^{n} \lambda_k c_k E[W_k]^m,$$

the ordering of the priority classes must follow the descending order of $\mu_i c_i$. Thus, the "μc" rule for determining the ordering of priority classes can be applied in general, for any cost functions of the form $\sum_{k=1}^{n} \lambda_k c_k E[W_k]^m$. However, unlike the linear cost case where fixed priority scheduling is optimal, in nonlinear cost functions, the optimal scheduling is usually either pure escalating priority scheduling or mixed priority scheduling (that is, a scheduling that is between no discrimination and maximum discrimination).

To conclude this section, we illustrate geometrically the solution for a two-class queueing model using Lagrange's method.

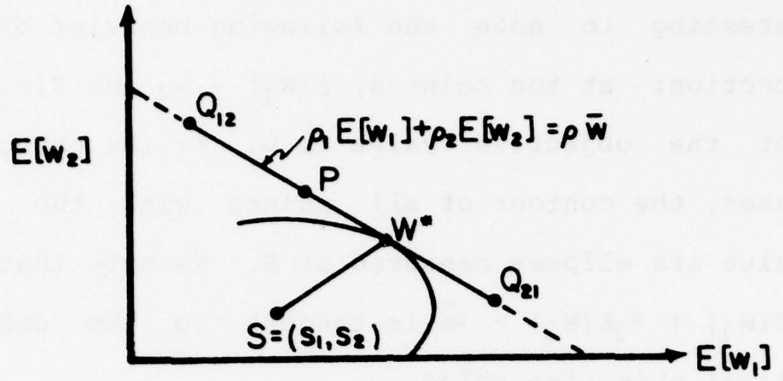


Figure 3.2 A Two-Class Queueing Model

Figure 3.2 shows a two-class queueing model with cost function defined by (3.2.1) for m=2 with positive s_1 and s_2. From (3.2.26), we have

$$E[W_1] = s_1 + \frac{\rho\overline{W} - \rho_1 s_1 - \rho_2 s_2}{(\mu_1 c_1)\left(\frac{\rho_1}{\mu_1 c_1} + \frac{\rho_2}{\mu_2 c_2}\right)},$$

$$\text{and } E[W_2] = s_2 + \frac{\rho\overline{W} - \rho_1 s_1 - \rho_2 s_2}{(\mu_2 c_2)\left(\frac{\rho_1}{\mu_1 c_1} + \frac{\rho_2}{\mu_2 c_2}\right)}.$$

Thus, if $W = (E[W_1], E[W_2])$ lies between Q_{12} and Q_{21}, then it is an optimal solution. However, if it does not, then we need to use the quadratic programming solution procedure that we described earlier to obtain a solution.

In this example, the point S is defined as (s_1, s_2). It is interesting to note the following behavior of the objective function: at the point S, $E[W_1] = s_1$ and $E[W_2] = s_2$, so that the objective value is 0. As the objective value increases, the contour of all points with the same objective value are elipses centered at S. We note that the line L: $\rho_1 E[W_1] + \rho_2 E[W_2] = \rho\overline{W}$ is tangent to the contour with the optimal objective value.

3.3 Examples

In this section, we illustrate the optimization techniques that were described in Sections 3.1 and 3.2 with a series of numerical examples using a 3-class priority queueing model with the following characteristics:

$$\begin{array}{lll}
\lambda_1 = 2 & \lambda_2 = 18/5 & \lambda_3 = 16/5 \\
E[S_1] = 1/10 & E[S_2] = 1/12 & E[S_3] = 1/8 \\
Var[S_1] = 1/240 & Var[S_2] = 1/180 & Var[S_3] = 1/320 \\
E[S_1^2] = 17/1200 & E[S_2^2] = 1/80 & E[S_3^2] = 3/160 \\
\mu_1 = 10 & \mu_2 = 12 & \mu_3 = 8 \\
\rho_1 = 0.2 & \rho_2 = 0.3 & \rho_3 = 0.4
\end{array}$$

And, $\rho = 9/10$, $\overline{W} = 2/3$.

The constraints (1.3.2) and (2.1.4) are:

(a) $0.2\ E[W_1] + 0.3\ E[W_2] + 0.4\ E[W_3] = 3/5$

(b.1) $0.2\ E[W_1] \geq 1/60$

(b.2) $0.3\ E[W_2] \geq 1/35$

(b.3) $0.4\ E[W_3] \geq 2/45$

(b.4) $0.2\ E[W_1] + 0.3\ E[W_2] \geq 1/15$

(b.5) $0.2\ E[W_1] + 0.4\ E[W_3] \geq 1/10$

(b.6) $0.3\ E[W_2] + 0.4\ E[W_3] \geq 7/45$

The space defined by the above constraints is shown in Figure 3.3.

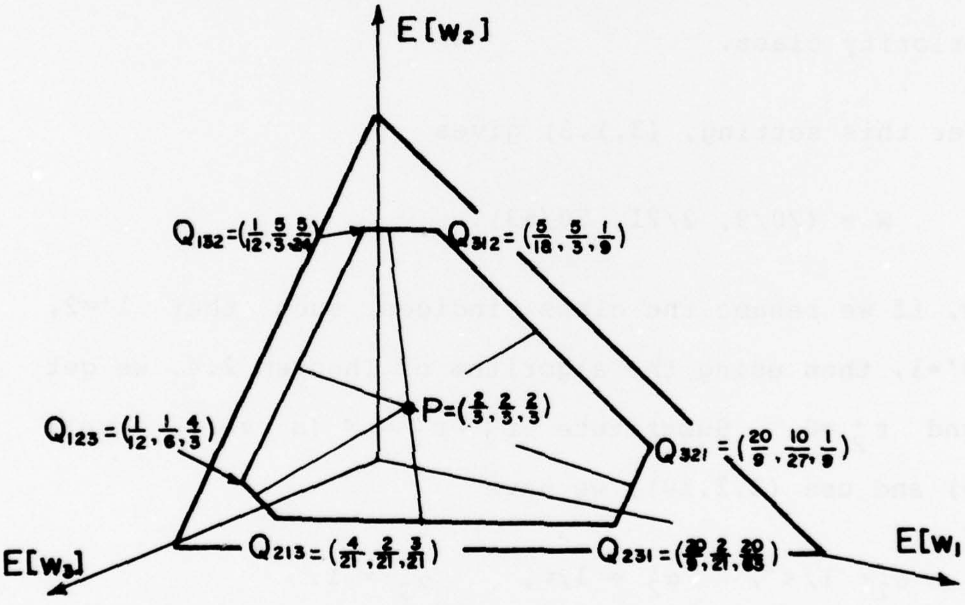


Figure 3.3 A 3-Class Priority Queueing Model

We now give the examples.

Example 1. Minimize $\sum_{k=1}^{n} \lambda_k c_k E[W_k]$

for $c_1 = 3, \qquad c_2 = 4, \qquad c_3 = 5.$

From the "μc"rule, we have

$$\mu_1 c_1 = 30, \quad \mu_2 c_2 = 48, \quad \mu_3 c_3 = 40.$$

Since $\mu_2 c_2 > \mu_3 c_3 > \mu_1 c_1$, the optimal scheduling is a fixed priority scheduling that sets class 2 as highest priority class, class 3 the next highest, and class 1 the lowest priority class.

Under this setting, (3.1.3) gives

$$W = (20/9,\ 2/21,\ 20/63).$$

Now, if we rename the class indices such that 1'=2, 2'=3, 3'=1, then using the algorithm of Theorem 2.4, we get $r'_1=0$ and $r'_2=0$. Substitute $r'_1=r'_2=\epsilon$ (a very small quantity) and use (2.2.10), we have

$$\alpha'_1 = 1/\epsilon^2, \quad \alpha'_2 = 1/\epsilon, \quad \alpha'_3 = 1.$$

Let $G=1/\epsilon$, then G is a very large number, and we have

$$\alpha_1 = 1, \quad \alpha_2 = G^2, \quad \alpha_3 = G.$$

This set of values can then be used in escalating priority scheduling to approximate the optimal fixed priority scheduling.

Example 2. Minimize $\sum_{k=1}^{n} \lambda_k c_k E[W_k]$

for $\quad c_1 = 4, \quad c_2 = 3, \quad c_3 = 5.$

In this case, we have

$$\mu_1 c_1 = 40, \qquad \mu_2 c_2 = 36, \qquad \mu_3 c_3 = 40.$$

Since $\mu_1 c_1 = \mu_3 c_3$, we have multiple optimal solutions. Among the extreme points, Q_{132} and Q_{312} are optimal solutions because $\mu_1 c_1 \geq \mu_3 c_3 > \mu_2 c_2$ and $\mu_3 c_3 \geq \mu_1 c_1 > \mu_2 c_2$.

The convex combination of these two points are the whole line segment $\overline{Q_{132}Q_{312}}$. Thus all points lying on the line segment $\overline{Q_{132}Q_{312}}$ are optimal. From (3.1.3) we have, $Q_{132} = (1/12, 5/3, 5/24)$ and $Q_{312} = (5/18, 5/3, 1/9)$. Thus, the optimal solutions are $kQ_{132} + (1-k)Q_{312}$, for $0 \leq k \leq 1$, or

$$W = ((5/18)-(7k/36),\ 5/3,\ (1/9)+(7k/72)).$$

One set of α_i's for these optimal solutions is

$$\alpha_1 = aG, \qquad \alpha_2 = 1, \qquad \alpha_3 = bG,$$

where G is a very large number and a, b are small positive quantities.

Example 3. Minimize max $\{\lambda_k c_k E[W_k];\ \ k=1,2,3\}$

for $\quad c_1 = 3, \quad c_2 = 4, \quad c_3 = 5.$

We first find a solution to the relaxed problem with constraint (a) only. From (3.1.10), we have

$$W = (24/19,\ 10/19,\ 9/19).$$

Since all the constraints (b.1) through (b,6) are satisfied, this solution is optimal.

Using the algorithm of Theorem 2.4, we get

$$\alpha_1 = 1, \qquad \alpha_2 = 2.8707, \qquad \alpha_3 = 3.2448.$$

Example 4. Minimize max $\{\lambda_k c_k E[W_k];\ k=1,2,3\}$

for $c_1 = 1, \qquad c_2 = 10, \qquad c_3 = 10.$

We first use (3.1.10) to find a solution to the relaxed problem with constraint (a) only. We have

$$W = (72/29,\ 4/29,\ 9/58).$$

Now, when we check the constraints (b.1) through (b.6), we see that (b.6) is violated. Therefore, we need to use the two-step procedure of solving n linear programming problems. Note that in this case, the solution must lie on the boundary.

According to the solution procedure, we have the following three optimization problems (the c constraints are from constraint set (3.1.5)):

(i) Minimize 2 $E[W_1]$ subject to (a), (b.1) ~ (b.6), and (c.1): $2\ E[W_1] \geq 36\ E[W_2]$,

(c.2): $2\ E[W_1] \geq 32\ E[W_3]$.

(ii) Minimize 36 $E[W_2]$ subject to (a), (b.1) ~ (b.6), and (c.3): $36\ E[W_2] \geq 2\ E[W_1]$,

(c.4): $36\ E[W_2] \geq 32\ E[W_3]$.

(iii) Minimize 32 $E[W_3]$ subject to (a), (b.1) ~ (b.6), and (c.5): $32\ E[W_3] \geq 2\ E[W_1]$,

(c.6): $32\ E[W_3] \geq 36\ E[W_2]$.

For the optimization problem (i), we find that the solution space is empty. So, we need only solve (ii) and (iii). *The solution spaces for these two optimization* problems are shown in Figure 3.4.

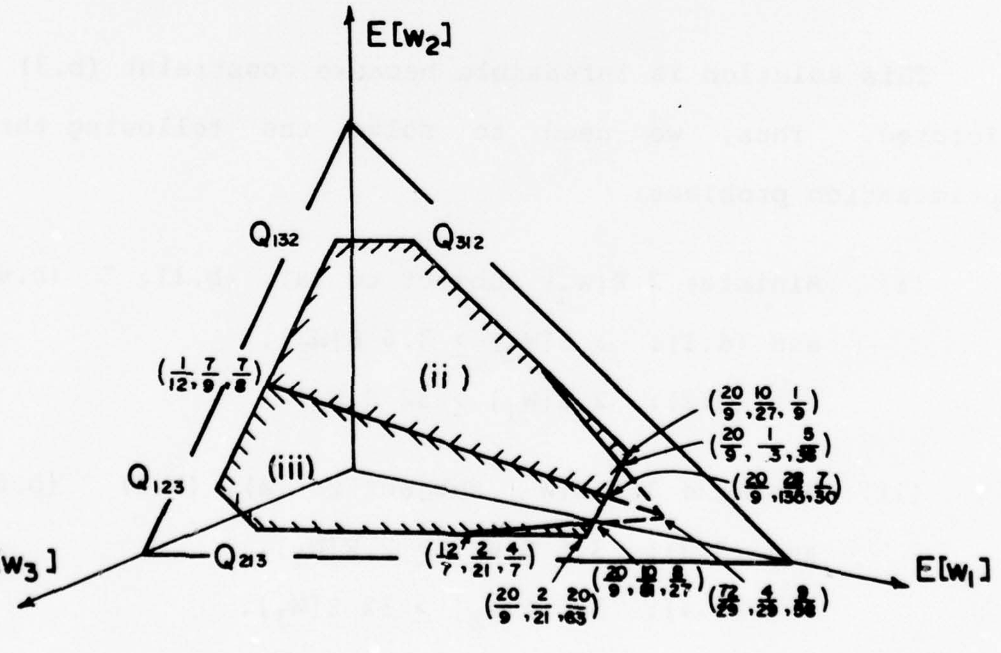

$E[W_2]$
Q_{132}
Q_{312}
(ii)
$(\frac{1}{12}, \frac{7}{9}, \frac{7}{8})$
$(\frac{20}{9}, \frac{10}{27}, \frac{1}{9})$
$(\frac{20}{9}, \frac{1}{3}, \frac{5}{36})$
(iii)
Q_{123}
$(\frac{20}{9}, \frac{28}{136}, \frac{7}{30})$
$E[W_3]$
Q_{213}
$(\frac{12}{7}, \frac{2}{21}, \frac{4}{7})$
$E[W_1]$
$(\frac{20}{9}, \frac{10}{81}, \frac{8}{27})$
$(\frac{20}{9}, \frac{2}{21}, \frac{20}{63})$
$(\frac{72}{25}, \frac{4}{25}, \frac{8}{75})$

Figure 3.4 Solution Spaces for Example 4

The optimal solution for both of these problems are

$$W = (20/9,\ 28/135,\ 7/30).$$

This is then the optimal solution to our original problem.

The procedure in Theorem 2.4 gives

$$\alpha_1 = 1, \qquad \alpha_2 = 1.1887G, \quad \alpha_3 = G.$$

Example 5. Minimize max $\{\lambda_k c_k E[W_k];\ k=1,2,3\}$

for $c_1 = 1$, $c_2 = 1$, $c_3 = 10$.

Again, we first obtain a solution to the relaxed problem with constraint (a) only. From (3.1.10), we have

$$W = (72/47,\ 40/47,\ 9/98).$$

This solution is infeasible because constraint (b.3) is violated. Thus, we need to solve the following three optimization problems:

(i) Minimize 2 $E[W_1]$ subject to (a), (b.1) ~ (b.6),
and (d.1): $2\,E[W_1] \geq 3.6\,E[W_2]$,
(d.2): $2\,E[W_1] \geq 32\,E[W_3]$.

(ii) Minimize 3.6 $E[W_2]$ subject to (a), (b.1) ~ (b.6),
and (d.3): $3.6\,E[W_2] \geq 2\,E[W_1]$,
(d.4): $3.6\,E[W_2] \geq 32\,E[W_3]$.

(iii) Minimize 32 $E[W_3]$ subject to (a), (b.1) ~ (b.6),

and (d.5): $32\ E[W_3] \geq 2\ E[W_1]$,

(d.6): $32\ E[W_3] \geq 3.6\ E[W_2]$.

The three regions for the above three optimizations are shown in Figure 3.5.

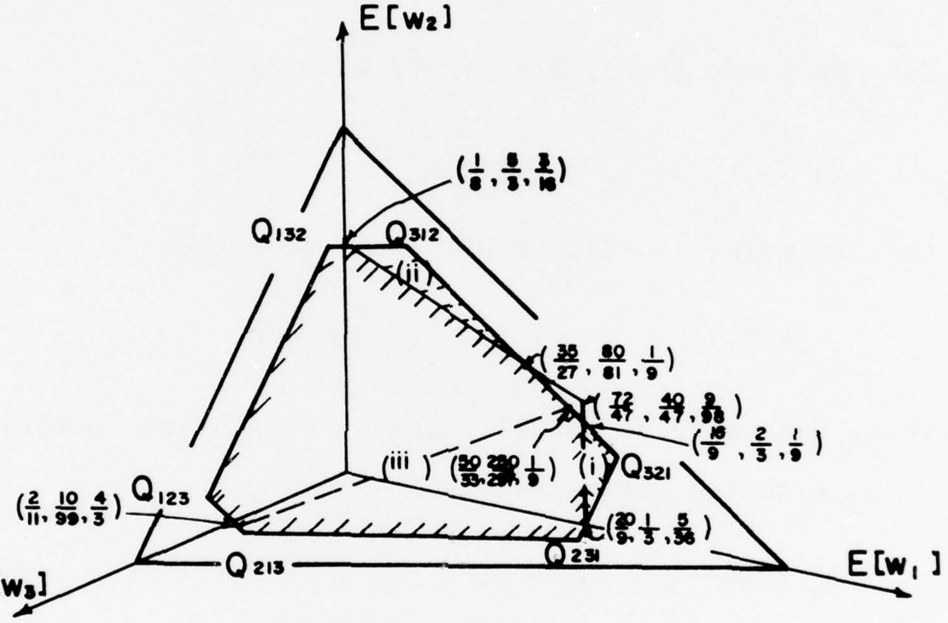


Figure 3.5 Solution Spaces for Example 5

For problem (i), the solution is

$$W = (16/9,\ 2/3,\ 1/9).$$

For problem (ii), the solution is

$$W = (35/27,\ 80/81,\ 1/9).$$

For problem (iii), we have multiple solutions as the whole line segment connecting (35/27, 80/81, 1/9) and (16/9, 2/3, 1/9).

Since all three optimization problems have the same objective value 32/9, all the solutions are optimal to our original problem. That is, our solution is the line segment connecting (35/27, 80/81, 1/9) and (16/9, 2/3, 1/9).

For the point (16/9, 2/3, 1/9), we obtain

$$\alpha_1 = 1, \qquad \alpha_2 = 4, \qquad \alpha_3 = G.$$

For the point (35/27, 80/81, 1/9), we obtain

$$\alpha_1 = 1, \qquad \alpha_2 = 1.4, \qquad \alpha_3 = G.$$

Thus, for any a, $1.4 \le a \le 4$, using escalating priority scheduling with the parameters as

$$\alpha_1 = 1, \qquad \alpha_2 = a, \qquad \alpha_3 = G,$$

will approximate the desired optimal performance.

Example 6. Minimize $\sum_{k=1}^{3} \lambda_k c_k E[W_k]^2$

for $\quad c_1 = 3, \qquad c_2 = 4, \qquad c_3 = 5.$

We first find a solution that satisfy constraint (a) only. From (3.2.26), we have

$$W = (48/29,\ 10/29,\ 12/29).$$

It can be seen that all the constraints (b.1) through (b.6) are satisfied. Thus, this is optimal to our original problem.

The procedure of Theorem 2.5 gives

$$\alpha_1 = 1, \qquad \alpha_2 = 7.7908, \qquad \alpha_3 = 6.2237.$$

Example 7. Minimize $\sum_{k=1}^{3} \lambda_k c_k E[W_k]^2$

for $\quad c_1 = 1, \qquad c_2 = 10, \qquad c_3 = 1.$

We first use (3.2.26) to obtain a solution to the relaxed problem with constraint (a) only. We have,

$$W = (24/29,\ 2/29,\ 30/29).$$

This solution is infeasible since constraint (b.2) is violated.

Thus, we need to use Kuhn-Tucker theorem to solve this problem. First, we add only the violated constraint (b.1) to constraint (a), and solve the maximization problem:

$$\text{Maximize } -\sum_{k=1}^{3} \lambda_k c_k E[W_k]^2.$$

The Kuhn-Tucker conditions are:

$$-2\lambda_1 c_1 E[W_1] - \rho_1 x_1 + \rho_2 x_2 + \rho_1 y_1' + \rho_1 y_3' + \rho_1 y_5' = -2\lambda_1 c_1 s_1,$$
$$-2\lambda_2 c_2 E[W_2] - \rho_1 x_1 + \rho_2 x_2 + \rho_2 y_2' + \rho_2 y_3' + \rho_2 y_6' = -2\lambda_2 c_2 s_2,$$
$$-2\lambda_3 c_3 E[W_3] - \rho_1 x_1 + \rho_2 x_2 + \rho_3 y_4' + \rho_3 y_5' + \rho_3 y_6' = -2\lambda_3 c_3 s_3,$$

$$\rho_1 E[W_1] + \rho_2 E[W_2] + \rho_3 E[W_3] = \rho \overline{W},$$
$$\rho_2 E[W_2] \geq 2/35.$$

Adding a slack variable T_1 to the last inequality constraint, then add R_1 through R_5 to each of the above constraints, by using artificial variable technique, we obtain

$$W = (40/49, 2/21, 50/49).$$

This solution satisfies all the constraints (b.1) through (b.6). Thus, this is the desired optimal solution.

For the values of α_i, we obtain

$$\alpha_1 = 1.3043, \quad \alpha_2 = G, \quad \alpha_3 = 1.$$

The above examples illustrate some of the objective functions that can be used in escalating priority scheduling. In particular, note that different objective functions with the same cost coefficients $\{c_i\}$ were used in examples 1, 3 and 6. Comparing Example 1 and Example 6, we see that when we change the cost function from linear to nonlinear, the optimal solution becomes an interior point which corresponds to an escalating priority scheduling. However, the ordering of priority is not changed. Example 3 is a completely different cost function and the ordering of priority for the optimal solution is different.

CHAPTER 4

BEHAVIOR OF ESCALATING PRIORITY SCHEDULING

4.0 Introduction

In this chapter, we discuss additional model behavior for multiclass M/G/1 queueing systems under escalating priority scheduling and make comparisons of the means and the variances of waiting times of escalating priority scheduling, fixed priority scheduling and FCFS scheduling. Without loss of generality, we will assume that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$.

In Section 4.1, we first investigate the bounds of mean waiting times for each priority class. Then we derive bounds on the ratios of mean waiting times of any two consecutive priority classes, and discuss the sensitivity of these ratios to the variation of system congestion. In Section 4.2 we discuss mean waiting times behavior at saturated congestion and show that the ratios of any two mean waiting times are inversely proportional to the ratios of the control parameters associated with them. In Section 4.3, we investigate the variances of waiting times for each class of jobs under various settings of the parameters $\{ \alpha_i \}$.

4.1 Behaviors of Mean Waiting Times

In this section, we discuss various behaviors of mean waiting times in escalating priority scheduling systems. We first derive bounds of mean waiting times for each priority class. Then we derive bounds on the ratios $E[W_{i+1}]/E[W_i]$ given a specific setting of parameters $\{\alpha_i\}$, and discuss the change of these ratios between the two bounds as a function of system congestion. This gives the sensitivity of waiting times to system fluctuations.

In Chapter 2, we showed that for any adjustable priority scheduling rule, mean waiting time for each priority class must be bounded by (2.1.1). Since escalating priority scheduling belongs to the category of adjustable priority scheduling rules, therefore (2.1.1) gives bounds of $E[W_i]$ for each class i. However, these bounds are derived under the assumption that each class can be assigned any arbitrary ordinal of priority. We will now study the bounds of mean waiting times given that a class must be assigned as the highest priority class, or the second highest priority class, etc.

These bounds are given by the following theorem:

THEOREM 4.1. In multiclass M/G/1 queueing systems under escalating priority scheduling, given $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$, we have

(4.1.1) $$(1-\rho)\overline{W}/(1-\sigma_i) \le E[W_i] \le \overline{W}/(1-\sigma_{i-1}).$$

<u>Proof:</u> (a) From Theorem 2.4, we have

$$E[W_1] \le E[W_2] \ldots \le E[W_n].$$

Thus, for $1 \le k \le i$, we have

$$\rho_k E[W_k] \le \rho_k E[W_i].$$

Therefore, $\sum\limits_{k=1}^{i} \rho_k E[W_k] \le \sum\limits_{k=1}^{i} \rho_k E[W_i] = (\sum\limits_{k=1}^{i} \rho_k) E[W_i] = \sigma_i E[W_i]$.

Now, from Theorem 2.2, we have

$$(1-\rho)\overline{W}\sigma_i/(1-\sigma_i) \le \sum_{k=1}^{i} \rho_k E[W_k].$$

Thus, $(1-\rho)\sigma_i \overline{W}/(1-\sigma_i) \le \sigma_i E[W_i]$

or, $(1-\rho)\overline{W}/(1-\sigma_i) \le E[W_i]$.

(b) From Theorem 2.4, we have

$$E[W_i] \le E[W_{i+1}] \le \ldots \le E[W_n].$$

Thus, for $i \le k \le n$, we have

$$\rho_k E[W_i] \le \rho_k E[W_k].$$

Therefore, $\sum\limits_{k=i}^{n} \rho_k E[W_i] = (\sum\limits_{k=i}^{n} \rho_k) E[W_i] \le \sum\limits_{k=i}^{n} \rho_k E[W_k]$.

But, from Theorem 2.2, we have

$$\sum_{k=i}^{n} \rho_k E[W_k] \leq \overline{W}(\sum_{k=i}^{n} \rho_k)/(1-\sigma_{i-1}).$$

Thus, $$(\sum_{k=i}^{n} \rho_k) E[W_i] \leq \overline{W}(\sum_{k=i}^{n} \rho_k)/(1-\sigma_{i-1})$$

or, $$E[W_i] \leq \overline{W}/(1-\sigma_{i-1}).$$

Figure 4.1 illustrates these bounds of mean waiting times for a 5-class queueing model. The mean waiting times for fixed priority scheduling and FCFS scheduling are represented as the dotted line and the broken line respectively.

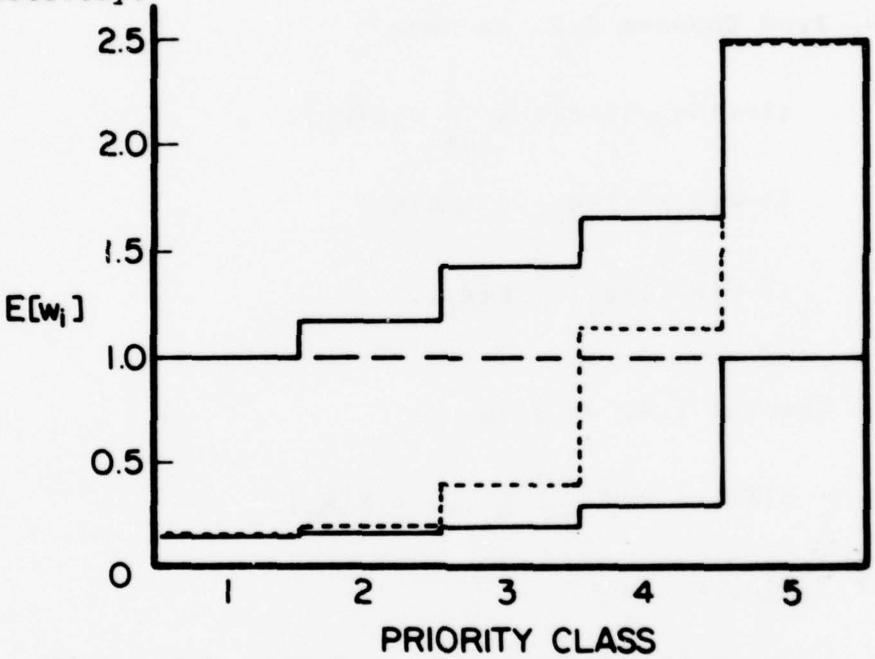


Figure 4.1 Bounds of Mean Waiting Times in a 5-Class Priority Queueing Model

We remark that in order that a set of mean waiting times $\{E[W_i]\}$ be feasible using escalating priority scheduling, they not only need to lie between the bounds as given by (4.1.1), but also need to satisfy the following:

(1.3.2) $$\rho_1 E[W_1] + \rho_2 E[W_2] + \ldots + \rho_n E[W_n] = \rho\overline{W},$$

and,

(2.2.2) $$E[W_1] \leq E[W_2] \leq \ldots \leq E[W_n].$$

Next, we derive the bounds of the ratio $E[W_{i+1}]/E[W_i]$ for each class i, given a specific setting of the parameters $\{\alpha_i\}$.

From Theorem 2.4, if we are given $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$, then using escalating priority scheduling, we have $E[W_1] \leq E[W_2] \leq \ldots \leq E[W_n]$. Thus, the lower bound of $E[W_{i+1}]/E[W_i]$ is

(4.1.2) $$E[W_{i+1}]/E[W_i] \geq 1. \qquad (1 \leq i \leq n-1)$$

The upper bound of $E[W_{i+1}]/E[W_i]$ can be obtained from the following theorem:

Theorem 4.2. In multiclass M/G/1 queueing systems under escalating priority scheduling, given $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$, we have

(4.1.3) $$\alpha_1 E[W_1] \geq \alpha_2 E[W_2] \geq \ldots \geq \alpha_n E[W_n].$$

<u>Proof:</u> From Theorem 1.5, we have

$$E[W_i] = \frac{\overline{W} - \sum\limits_{k=i+1}^{n} (1 - \frac{\alpha_k}{\alpha_i}) \rho_k E[W_k]}{1 - \sum\limits_{k=1}^{i-1} (1 - \frac{\alpha_i}{\alpha_k}) \rho_k} \, .$$

Let the numerator and denominator of the above expression be N_i and D_i respectively. Then,

$$N_i = \overline{W} - \sum_{k=i+1}^{n} \rho_k E[W_k] + \frac{1}{\alpha_i} \sum_{k=i+1}^{n} \alpha_k \rho_k E[W_k] \, ,$$

$$N_{i+1} = \overline{W} - \sum_{k=i+2}^{n} \rho_k E[W_k] + \frac{1}{\alpha_{i+1}} \sum_{k=i+2}^{n} \alpha_k \rho_k E[W_k]$$

$$= \overline{W} - \sum_{k=i+1}^{n} \rho_k E[W_k] + \frac{1}{\alpha_{i+1}} \sum_{k=i+1}^{n} \alpha_k \rho_k E[W_k] \, ,$$

$$D_i = 1 - \sum_{k=1}^{i} \rho_k + \alpha_{i+1} \sum_{k=1}^{i} \frac{\rho_k}{\alpha_k} \, ,$$

$$= 1 - \sigma_i + \alpha_{i+1} \sum_{k=1}^{i} \frac{\rho_k}{\alpha_k} \, ,$$

$$D_{i+1} = 1 - \sigma_{i+1} + \alpha_i \sum_{k=1}^{i-1} \frac{\rho_k}{\alpha_k}$$

$$= 1 - \sigma_i + \alpha_i \sum_{k=1}^{i} \frac{\rho_k}{\alpha_k} \, .$$

So, $\alpha_i E[W_i] - \alpha_{i+1} E[W_{i+1}]$

$$= \frac{\alpha_i N_i}{D_i} - \frac{\alpha_{i+1} N_{i+1}}{D_{i+1}}$$

$$= \frac{\alpha_i D_{i+1} N_i - \alpha_{i+1} D_i N_{i+1}}{D_i D_{i+1}} \, .$$

The numerator $\alpha_i D_{i+1} N_i - \alpha_{i+1} D_i N_{i+1}$

$$= (\alpha_i(1-\sigma_i) + \alpha_i\alpha_{i+1}\sum_{k=1}^{i}\frac{\rho_k}{\alpha_k})$$

$$\times (\overline{W} - \sum_{k=i+1}^{n}\rho_k E[W_k] + \frac{1}{\alpha_i}\sum_{k=i+1}^{n}\alpha_k\rho_k E[W_k])$$

$$- (\alpha_{i+1}(1-\sigma_i) + \alpha_i\alpha_{i+1}\sum_{k=1}^{i}\frac{\rho_k}{\alpha_k})$$

$$\times (\overline{W} - \sum_{k=i+1}^{n}\rho_k E[W_k] + \frac{1}{\alpha_{i+1}}\sum_{k=i+1}^{n}\alpha_k\rho_k E[W_k])$$

$$= (\alpha_i-\alpha_{i+1})\{(1-\sigma_i)(\overline{W} - \sum_{k=i+1}^{n}\rho_k E[W_k])$$

$$- (\sum_{k=1}^{i}\frac{\rho_k}{\alpha_k})(\sum_{k=i+1}^{n}\alpha_k\rho_k E[W_k])\}$$

$$= (\alpha_i-\alpha_{i+1})\{(1-\sigma_i)\overline{W}$$

$$- \sum_{k=i+1}^{n}\rho_k E[W_k][(1-\sigma_i) + \sum_{j=1}^{i}\frac{\alpha_k}{\alpha_j}\rho_j]\}$$

$$= (\alpha_i-\alpha_{i+1})\{(1-\sigma_i)\overline{W}$$

$$- \sum_{k=i+1}^{n}\rho_k E[W_k][1 - \sum_{j=1}^{i}(1-\frac{\alpha_k}{\alpha_j})\rho_j]\}$$

$$= (\alpha_i-\alpha_{i+1})\{(1-\sigma_i)\overline{W} - \sum_{k=i+1}^{n}\rho_k E[W_k]$$

$$\times [1 - \sum_{j=1}^{k-1}(1-\frac{\alpha_k}{\alpha_j})\rho_j + \sum_{j=i+1}^{k-1}(1-\frac{\alpha_k}{\alpha_j})\rho_j]\}$$

$$= (\alpha_i - \alpha_{i+1})\{(1-\sigma_i)\overline{W} - \sum_{k=i+1}^{n} \rho_k E[W_k] D_k$$

$$- \sum_{k=i+1}^{n} \rho_k E[W_k] \sum_{j=i+1}^{k-1} (1 - \frac{\alpha_k}{\alpha_j}) \rho_j \}$$

$$= (\alpha_i - \alpha_{i+1})\{(1-\sigma_i)\overline{W} - \sum_{k=i+1}^{n} \rho_k \frac{N_k}{D_k} D_k$$

$$- \sum_{k=i+1}^{n} \sum_{j=i+1}^{k-1} (1 - \frac{\alpha_k}{\alpha_j}) \rho_j \rho_k E[W_k] \}$$

$$= (\alpha_i - \alpha_{i+1})\{(1-\sigma_i)\overline{W} - \sum_{k=i+1}^{n} \rho_k N_k$$

$$- \sum_{k=i+2}^{n} \sum_{j=i+1}^{k-1} (1 - \frac{\alpha_k}{\alpha_j}) \rho_j \rho_k E[W_k] \}$$

$$= (\alpha_i - \alpha_{i+1})\{(1-\sigma_i)\overline{W} - \sum_{k=i+1}^{n} \rho_k (\overline{W} - \sum_{j=k+1}^{n} (1 - \frac{\alpha_j}{\alpha_k}) \rho_j E[W_j]$$

$$- \sum_{j=i+1}^{n} \sum_{k=j-1}^{n} (1 - \frac{\alpha_k}{\alpha_j}) \rho_j \rho_k E[W_k] \}$$

$$= (\alpha_i - \alpha_{i+1})\{(1-\sigma_i)\overline{W} - (\sum_{k=i+1}^{n} \rho_k)\overline{W}$$

$$+ \sum_{k=i+1}^{n} \sum_{j=k+1}^{n} (1 - \frac{\alpha_j}{\alpha_k}) \rho_k \rho_j E[W_j]$$

$$- \sum_{j=i+1}^{n} \sum_{k=j-1}^{n} (1 - \frac{\alpha_k}{\alpha_j}) \rho_j \rho_k E[W_k]$$

$$= (\alpha_i - \alpha_{i+1})\{(1-\sigma_i - \sum_{k=i+1}^{n} \rho_k)\overline{W}\}$$

$$= (\alpha_i - \alpha_{i+1})(1-\rho)\overline{W} .$$

Thus, $\alpha_i E[W_i] - \alpha_{i+1} E[W_{i+1}] = (\alpha_i - \alpha_{i+1})(1-\rho)\overline{W}/D_i D_{i+1} \geq 0$.

From (4.1.2) and (4.1.3), we have the following bounds:

$$(4.1.4) \qquad 1 \leq E[W_{i+1}]/E[W_i] \leq \alpha_i/\alpha_{i+1}. \qquad (1 \leq i \leq n-1)$$

Note that these bounds are independent of the congestion factor ρ.

We next examine the change of these ratios of mean waiting times as ρ increases from 0 to 1. We will assume that the arrivals from each class increase proportionately so that λ_i/λ is fixed for all i. Under this condition,

$$(4.1.5) \qquad p_i = \rho_i/\rho$$

is fixed.

The ratios of $E[W_{i+1}]/E[W_i]$ can be obtained by first examining the ratios $E[W_i]/\bar{W}$:

From (1.3.10), we have

$$E[W_i]/\bar{W} = \frac{1 - \sum_{k=i+1}^{n} (1 - \frac{\alpha_k}{\alpha_i}) \rho_k (E[W_k]/\bar{W})}{1 - \sum_{k=1}^{i-1} (1 - \frac{\alpha_i}{\alpha_k}) \rho_k} .$$

Substituting (4.1.5) into the above equation, we obtain

$$(4.1.6) \qquad E[W_i]/\bar{W} = \frac{1 - (\sum_{k=i+1}^{n} (1 - \frac{\alpha_k}{\alpha_i}) p_k (E[W_k]/\bar{W}))\rho}{1 - (\sum_{k=1}^{i-1} (1 - \frac{\alpha_i}{\alpha_k}) p_k)\rho} ,$$

which is a function of ρ only.

Therefore, given fixed ratios of arrivals among different priority classes, (4.1.6) can be used recursively (from n backwards to 1) to calculate $E[W_i]/\overline{W}$ for various congestion rates.

To illuatrate this, we consider a 5-class queueing model with the following characteristics: $E[S_i] = 1$, $E[S_i^2] = 2$, $\lambda_i = \lambda/5$, $\alpha_i = 1/i$, and $p_i = 1/5$, for $1 \leq i \leq 5$.

Figure 4.2 depicts the change of each $E[Wi]/\overline{W}$ in the range $0 < \rho < 1$. The gradual slopes of these curves show that for small fluctuations in congestion, the ratios of waiting times between any two priority classes will not change much. Even over a rather large range of ρ, the mean waiting time ratios do not have a large change.

Since $\overline{W}$ itself is an increasing function of ρ, we note that as ρ approaches 1, each $E[W_i]$ will approach infinity. The mean waiting times $E[W_i]$ and $\overline{W}$ are shown in Figure 4.3.

For comparison, we show the corresponding measures of the same system operated under fixed priority scheduling in Figures 4.4 and 4.5.

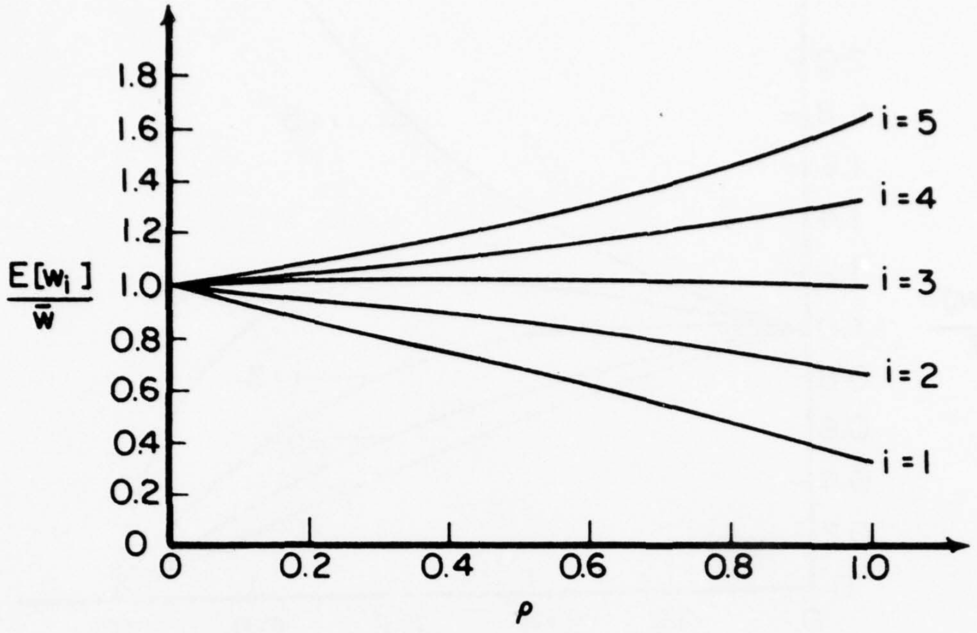


Figure 4.2 $E[W_i]/\bar{W}$ vs System Congestion in Escalating priority Scheduling

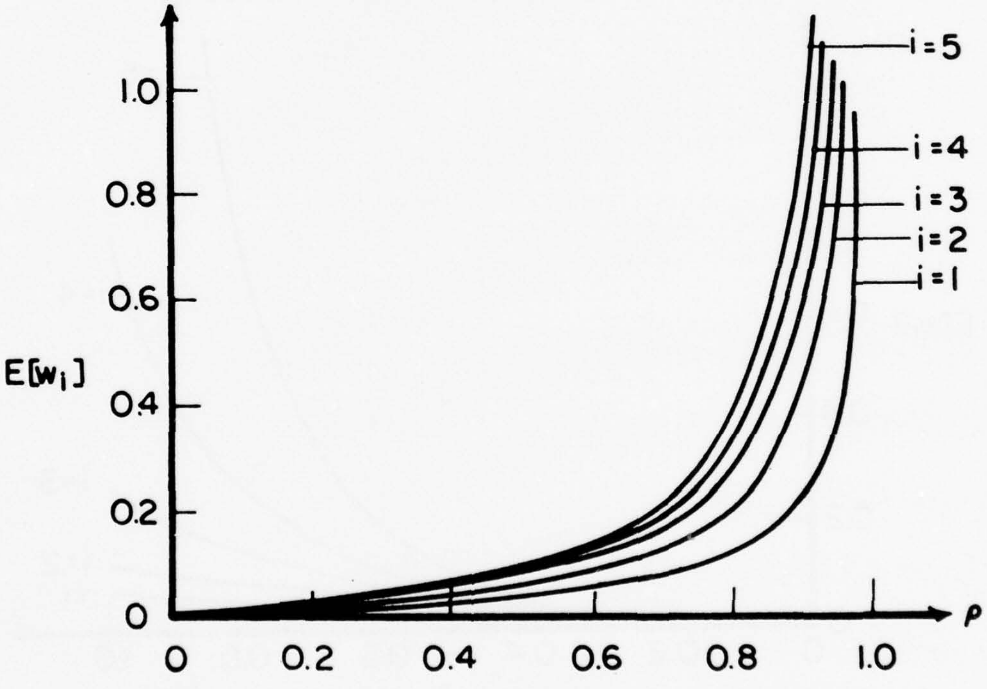


Figure 4.3 $E[W_i]$ vs System Congestion in Escalating priority Scheduling

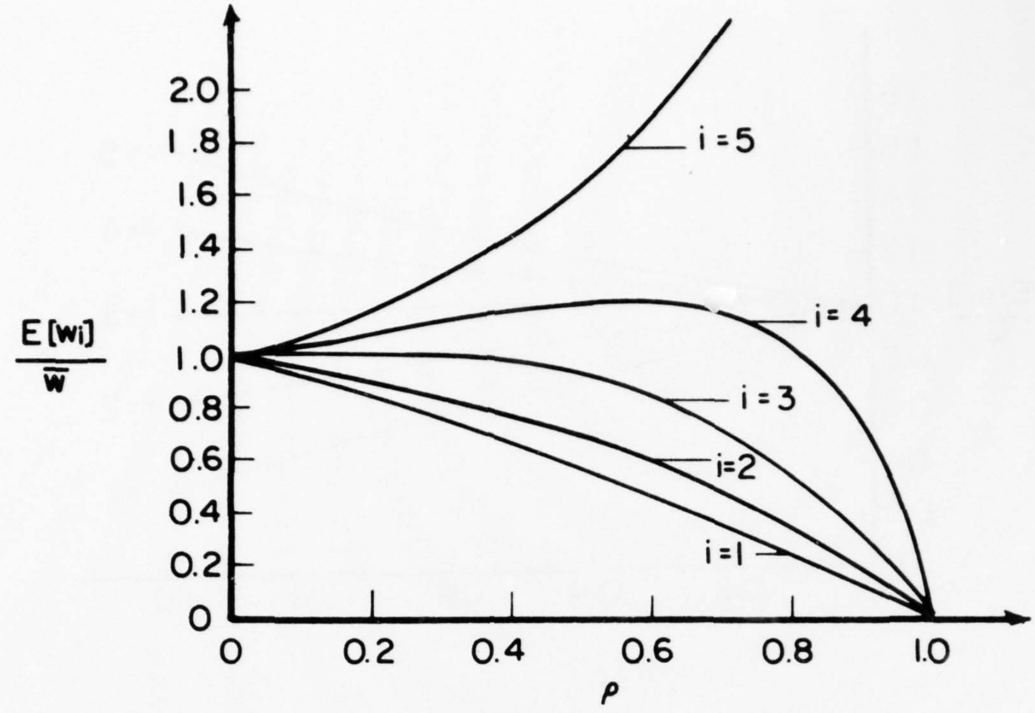


Figure 4.4 $E[W_i]/\overline{W}$ vs System Congestion in Fixed priority Scheduling

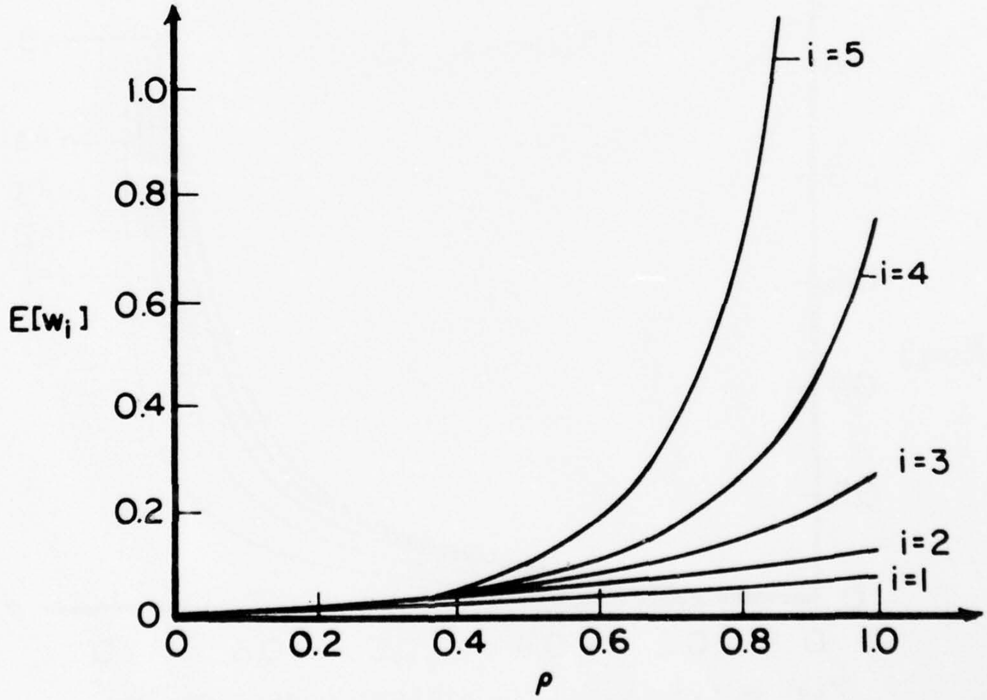


Figure 4.5 $E[W_i]$ vs System Congestion in Fixed priority Scheduling

The mean waiting times behaviors of escalating priority scheduling in Figures 4.2 and 4.3 show that as system congestion increases, every priority class has a significant increase in mean waiting time and that strong couplings can occur between priority classes. However, in fixed priority scheduling, increased congestion has almost no effect on the mean waiting times for higher priority classes, but a large effect on the lowest priority class as can be seen in Figures 4.4 and 4.5.

We remark that the strength of couplings between priority classes in escalating priority scheduling depends on the values of the parameters $\{ \alpha_i \}$. As the ratio α_i/α_{i+1} increases, the strength of coupling between classes i and i+1 decreases. In the limiting situation when $\alpha_i / \alpha_{i+1} \rightarrow \infty$, this coupling effect disappears, and we have a MEFP scheduling. This mixed priority scheduling rule has characteristics of both fixed priority and escalating priority scheduling: It is adjustable so that relative waiting times among priority classes can be controlled and it can be operated at saturated congestion while still giving finite mean waiting times to the higher priority classes. This property of MEFP scheduling rule was first observed by Kleinrock [KLEI66].

4.2 Waiting Times Behavior at Saturated Congestion

In this section, we investigate mean waiting times behavior at saturated congestion -- that is, $\rho \simeq 1$ but remains strictly less than one, preserving stability. We first study the mean waiting times behavior of fixed priority scheduling. From Theorem 1.3, we have

$$(1.3.4) \qquad E[W_i]_{FP} = (1-\rho)\overline{W}/(1-\sigma_{i-1})(1-\sigma_i). \qquad (1\leq i\leq n)$$

From (1.3.1), $\overline{W}$ is defined as

$$(1.3.1) \qquad \overline{W} = (\sum_{k=1}^{n} \lambda_k E[S_k^2])/2(1-\rho).$$

Thus, $$E[W_i]_{FP} = (\sum_{k=1}^{n} \lambda_k E[S_k^2])/2(1-\sigma_{i-1})(1-\sigma_i). \qquad (1\leq i\leq n)$$

Therefore, for $1\leq i\leq n-1$,

$$\lim_{\rho\to 1} E[W_i]_{FP} < \infty,$$

while for i=n, we have

$$\lim_{\rho\to 1} E[W_n]_{FP} = \infty.$$

Therefore, in fixed priority scheduling with saturated traffic, higher priority classes will have finite mean waiting times and the lowest priority class will have an unbounded mean waiting time. This behavior is reflected in Figure 4.4 where $\lim_{\rho\to 1} E[W_i]/\overline{W} = 0$ for $i<n$ and $\lim_{\rho\to 1} E[W_n]/\overline{W} > 0$.

Next, we examine the ratios among the $E[W_i]$'s under escalating priority scheduling. These ratios can be obtained from the ratios $\{E[W_i]/\overline{W}\}$ so we will first study the behavior of these ratios. Practically, $E[W_i]/\overline{W}$ is a measure of the degree of discrimination applied to class i by the underlying scheduling relative to an indiscriminated treatment.

The following theorem gives the ratios $\{E[W_i]/\overline{W}\}$ when escalating priority scheduling is operated under saturated congestion.

THEOREM 4.3. In multiclass M/G/1 queueing systems under escalating priority scheduling, in the limiting situation when $\rho \simeq 1$, we have

$$\lim_{\rho\to 1} \alpha_i E[W_i]/\overline{W} = 1/\sum_{k=1}^{n} (\rho_k/\alpha_k), \qquad (1\leq i\leq n) \tag{4.2.1}$$

which is a constant independent of the class.

Proof: Given a set of finite α_i's and $\rho = \rho_1 + \rho_2 + \ldots + \rho_n \simeq 1$, we let

$$d = \sum_{k=1}^{n} \frac{\rho_k}{\alpha_k} .$$

Then, d is a finite constant.

We will prove (4.2.1) by induction (from n backwards to 1).

First, from (1.3.10), we have

$$\lim_{\rho\to 1} \frac{E[W_n]}{\overline{W}} = 1 \,/\, \{1 - \sum_{k=1}^{n-1} (1 - \frac{\alpha_n}{\alpha_k}) \rho_k\}$$

$$= 1 \,/\, \{1 - \sum_{k=1}^{n-1} \rho_k + \alpha_n \sum_{k=1}^{n-1} \frac{\rho_k}{\alpha_k}\}$$

$$= 1 \,/\, \{1 - \sum_{k=1}^{n-1} \rho_k + \alpha_n (d - \frac{\rho_n}{\alpha_n})\}$$

$$= 1 \,/\, \{1 - \rho + \alpha_n d\}$$

$$= 1 \,/\, \alpha_n d \,.$$

Hence, $\lim_{\rho\to 1} \frac{\alpha_n E[W_n]}{\overline{W}} = \frac{1}{d}$.

Now, suppose (4.2.1) is true for $k=i+1, i+2, \ldots, n$, then, again from (1.3.10),

$$\lim_{\rho\to 1} \frac{E[W_i]}{\overline{W}} = \frac{1 - \sum_{k=i+1}^{n} (1 - \frac{\alpha_k}{\alpha_i}) \rho_k E[W_k] \,/\, \overline{W}}{1 - \sum_{k=1}^{i-1} (1 - \frac{\alpha_i}{\alpha_k}) \rho_k}$$

$$= \frac{1 - (\sum_{k=i+1}^{n} \rho_k E[W_k] \,/\, \overline{W}) + \frac{1}{\alpha_i} (\sum_{k=i+1}^{n} \alpha_k \rho_k E[W_k] \,/\, \overline{W})}{1 - \sum_{k=1}^{i-1} \rho_k + \alpha_i \sum_{k=1}^{i-1} \frac{\rho_k}{\alpha_k}}$$

$$= \frac{1 - (\sum_{k=i+1}^{n} \frac{\rho_k}{\alpha_k d}) + \frac{1}{\alpha_i} (\sum_{k=i+1}^{n} \frac{\rho_k}{d})}{1 - \sigma_{i-1} + \alpha_i (d - \sum_{k=i}^{n} \frac{\rho_k}{\alpha_k})}$$

$$= \frac{1 - \frac{1}{d}(\sum_{k=i+1}^{n} \frac{\rho_k}{\alpha_k}) + \frac{1}{\alpha_i d} \sum_{k=i+1}^{n} \rho_k}{1 - \sigma_{i-1} + \alpha_i d - \alpha_i (\sum_{k=i}^{n} \frac{\rho_k}{\alpha_k})}$$

$$= \frac{\frac{1}{\alpha_i d}\{\alpha_i d - \alpha_i (\sum_{k=i+1}^{n} \frac{\rho_k}{\alpha_k}) + \sum_{k=i+1}^{n} \rho_k\}}{1 - \sigma_{i-1} + \alpha_i d - \rho_i - \alpha_i (\sum_{k=i+1}^{n} \frac{\rho_k}{\alpha_k})}$$

$$= \frac{1}{\alpha_i d} \cdot \frac{(\rho - \sigma_i) + \alpha_i d - \alpha_i (\sum_{k=i+1}^{n} \frac{\rho_k}{\alpha_k})}{1 - \sigma_i + \alpha_i d - \alpha_i (\sum_{k=i+1}^{n} \frac{\rho_k}{\alpha_k})}$$

$$= \frac{1}{\alpha_i d} \, .$$

Hence, we have

$$\lim_{\rho \to 1} \frac{\alpha_i E[W_i]}{\overline{W}} = \frac{1}{d} \, .$$

This completes the proof.

As an immediate consequence, we have the following:

(4.2.2) $$\lim_{\rho\to 1} E[W_i]/E[W_j] = \alpha_j/\alpha_i. \qquad (1\le i\le n,\ 1\le j\le n)$$

Stated in another way, we have

(4.2.3) $$\lim_{\rho\to 1} E[W_1]:E[W_2]:\ldots:E[W_n] = \frac{1}{\alpha_1}:\frac{1}{\alpha_2}:\ldots:\frac{1}{\alpha_n}.$$

This result attaches a significant meaning to the parameters $\{\alpha_i\}$: They are "discrimination factors" that set decisive measure to the ratios of mean waiting times among the priority classes. Under saturated congestion, the reciprocals of the relative ratios of these discrimination factors are the relative ratios of waiting times that can be expected.

We observe the following: In Section 3.2, we showed that the optimal solution to the objective function

(3.2.29) $$\text{Minimize } \sum_{k=1}^{n} \lambda_k c_k E[W_k]^2$$

with only the constraint (1.3.2) must satisfy

(3.2.27) $$E[W_1]:E[W_2]:\ldots:E[W_n] = \frac{1}{\mu_1 c_1}:\frac{1}{\mu_2 c_2}:\ldots:\frac{1}{\mu_n c_n}.$$

If we now examine the constraint set (2.1.4) at saturated congestion, we can see that for each inequality

$$\sum_{k\in I} \rho_k E[W_k] \ge (1-\rho)(\sum_{k\in I} \rho_k)\overline{W}/(1-\sum_{k\in I} \rho_k),$$

the right hand side vanishes when $\rho \to 1$.

Thus, in this limiting situation, (2.1.4) becomes ineffective and every solution satisfying (1.3.2) is optimal.

Comparing (3.2.27) with (4.2.3), we see that if we set

$$(4.2.4) \qquad \alpha_i = \mu_i c_i, \qquad (1 \leq i \leq n)$$

then at saturated congestion, an escalating priority scheduling will be optimal for the objective function (3.2.29), when the parameters $\{ \alpha_i \}$ are set equal to the values in (4.2.4). Therefore, when the system congestion is heavy, (i.e., when ρ is close to 1), (4.2.4) can be used to give an approximate optimal solution to (3.2.29).

4.3 Variance of Waiting Times

In this section, we investigate the variances of waiting times for escalating priority scheduling and compare them with the variances of FCFS and fixed priority scheduling.

We first give the following theorem on the variances of waiting times for FCFS and fixed priority schedulings (c.f., [TAKA64]).

THEOREM 4.4. In multiclass M/G/1 queueing systems with $\rho < 1$, we have the following:

(a) Under FCFS scheduling, the variance of waiting times for all classes is

$$\text{(4.3.1)} \qquad \mathrm{Var}[W_{FCFS}] = \overline{W}^2 + \Big(\sum_{k=1}^{n} \lambda_k\, E[S_k^3]\Big) \,/\, 3\,(1-\rho).$$

(b) Under fixed priority scheduling, the variance of waiting times for priority class i $(1 \leq i \leq n)$ is

$$\text{(4.3.2)} \qquad \mathrm{Var}[Wi]_{FP} = \frac{\sum_{k=1}^{n} \lambda_k\, E[S_k^3]}{3(1-\sigma_{i-1})^2(1-\sigma_i)} + \frac{\overline{W}(1-\rho)\Big(\sum_{k=1}^{i-1} \lambda_k\, E[S_k^2]\Big)}{(1-\sigma_{i-1})^3(1-\sigma_i)} + \frac{\overline{W}(1-\rho)\Big\{\sum_{k=1}^{i} \lambda_k\, E[S_k^2] - \overline{W}(1-\rho)\Big\}}{(1-\sigma_{i-1})^2(1-\sigma_i)^2}.$$

As a corollary, we have

$$\text{(4.3.3)} \qquad \mathrm{Var}[W_1]_{FP} < \mathrm{Var}[W_2]_{FP} < \ldots < \mathrm{Var}[W_n]_{FP}.$$

From the above theorem, we note the following: First, under FCFS scheduling, we have

$$(4.3.4) \qquad \mathrm{Var}[Wi]_{FCFS} \;/\; E[Wi]^2_{FCFS}$$

$$= \mathrm{Var}[Wi]_{FCFS} \;/\; \overline{W}^2$$

$$= 1 + (\sum_{k=1}^{n} \lambda_k\, E[S_k^3])/\; 3\, \overline{W}^2\ (1-\rho)$$

$$> 1.$$

Thus, the coefficient of variation (Cv) of the waiting time variable W_{FCFS}, defined as the square root of $\mathrm{Var}[W_{FCFS}]/E[W_{FCFS}]^2$, is greater than one.

As a reference, the coefficient of variation of an exponentially distributed random variable is one. Thus, the density function of the waiting time variable W_{FCFS} is flatter than an exponential function, and has a "longer tail".

Second, for fixed priority scheduling, using (4.3.2), the variance of the lowest priority class is

$$(4.3.5)\ \mathrm{Var}[W_n]_{FP} = \frac{\sum_{k=1}^{n} \lambda_k\, E[S_k^3]}{3(1-\sigma_{n-1})^2(1-\rho)}$$

$$+ \frac{\overline{W}\ \sum_{k=1}^{n-1} \lambda_k\, E[S_k^2]}{(1-\sigma_{n-1})^3} + \frac{\overline{W}^2}{(1-\sigma_{n-1})^2}\ .$$

From Theorem 1.3, we have

$$(1.3.6) \qquad E[W_n]_{FP} = \overline{W} \,/\, (1-\sigma_{n-1}).$$

Thus,

$$(4.3.6) \qquad \text{Var}[W_n]_{FP} \,/\, E[W_n]^2_{FP}$$

$$= 1 + \frac{\sum_{k=1}^{n} \lambda_k E[S_k^3]}{3\overline{W}^2(1-\rho)} + \frac{\sum_{k=1}^{n-1} \lambda_k E[S_k^2]}{\overline{W}(1-\sigma_{n-1})}$$

$$> \text{Var}[W_{FCFS}]/E[W_{FCFS}]^2 \,.$$

Therefore, for the lowest priority class, not only is $\text{Var}[Wn]_{FP}$ greater than $\text{Var}[W_{FCFS}]$, but the coefficient of variation (which reflects the relative dispersion of waiting times) is also greater than the coefficient of variation of waiting when FCFS scheduling is used.

Third, for the highest priority class, (4.3.2) gives

$$(4.3.7) \quad \text{Var}[W_1]_{FP} = \frac{\sum_{k=1}^{n} \lambda_k E[S_k^3]}{3(1-\rho_1)} + \frac{\overline{W}(1-\rho)\lambda_1 E[S_1^2]}{(1-\rho_1)^2} + \frac{\overline{W}^2(1-\rho)^2}{(1-\rho_1)^2} \,.$$

Thus, $\text{Var}[W_{FCFS}] - \text{Var}[W_1]_{FP}$

$$= \overline{W}^2 + \frac{\sum_{k=1}^{n} \lambda_k E[S_k^3]}{3(1-\rho)} - \frac{\sum_{k=1}^{n} \lambda_k E[S_k^3]}{3(1-\rho_1)}$$

$$+ \frac{\overline{W}(1-\rho)\{\overline{W}(1-\rho)-\lambda_1 E[S_1^2]\}}{(1-\rho_1)^2}$$

$$> \overline{W}^2 + \overline{W}(1-\rho)\{\tfrac{1}{2} \sum_{k=1}^{n} \lambda_k E[S_k^2] - \lambda_1 E[S_1^2]\} \,/\, (1-\rho_1)^2$$

$$> \overline{W}^2 - E[W_1]\{\frac{1}{2} \sum_{k=1}^{n} \lambda_k E[S_k^2]\} / (1-\rho_1)$$

$$= \overline{W} - E[W_1]^2_{FP}$$

$$> 0.$$

Therefore, by giving top preference to a class, not only will the mean waiting time for jobs in this class decrease, the variance of waiting will also decrease. As to the coefficient of variation of waiting times for this class, no general conclusion can be drawn.

In escalating priority scheduling, we know that the control parameters can be used to adjust the preferences given to each class and the stronger the discrimination applied, the more dispersion between waiting times of different priority classes there is. We also know that the mean waiting times for jobs that belong to the higher priority classes will decrease as the preferences given to these classes increase. Similarly, for lower priority classes, the increase of discrimination applied to them will increase their mean waiting times. It is desirable to also understand how changes in discriminations affect the variance of waiting times for jobs within each priority class.

To investigate the variance behavior of escalating priority scheduling, a simulation model of a 3-class priority queueing system was developed. The details of the simulation is given in Appendix A. In the following, we present the results of the variance behavior from the simulation experiments.

In Figure 4.6, the theoretical mean waiting times of the model under different levels of discrimination in escalating priority scheduling are shown. For this 3-class priority scheuling model, we have two degrees of freedom to adjust the parameters. Let $r_1 = \alpha_2/\alpha_1$ and $r_2 = \alpha_3/\alpha_2$ be the ratios of the control parameters. Recalling that we have restricted the $\{\alpha_i\}$ to be such that $\alpha_1 \geq \alpha_2 \geq \alpha_3$, we therefore have $0 < r_i \leq 1$ for $1 \leq i \leq 2$. The three surfaces above the feasible values of (r_1, r_2) are the mean waiting times for the three different classes. The waiting times performance corresponding to either r_i being zero -- which represent the behavior of MEFP scheduling -- is added to make the surfaces closed.

For this model, if we only consider the parametric change $(\alpha_1, \alpha_2, \alpha_3) = (a/k, a, ak)$ for $k: 1 \rightarrow 0$ (a is an arbitrary positive constant), then $r_1 = r_2 = k$ and $0 < k \leq 1$. The mean waiting times under this change of discriminations are shown in Figure 4.7.

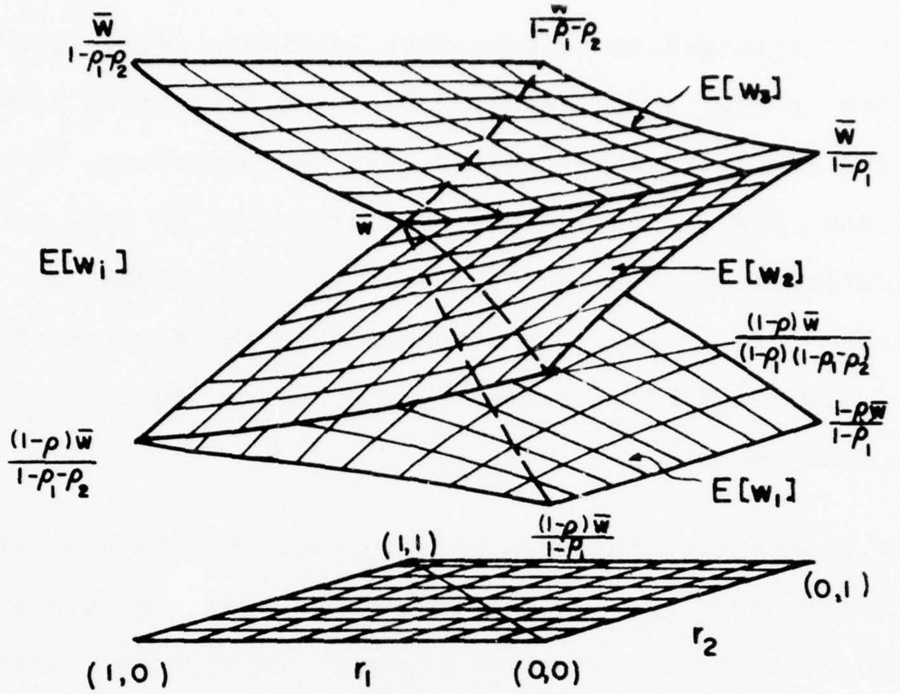

$\frac{\bar{w}}{1-\rho_1-\rho_2}$
$\frac{\bar{w}}{1-\rho_1-\rho_2}$
E[w3]
$\frac{\bar{w}}{1-\rho_1}$
$\bar{w}$
E[wi]
E[w2]
$\frac{(1-\rho)\bar{w}}{(1-\rho_1)(1-\rho_1-\rho_2)}$
$\frac{1-\rho\bar{w}}{1-\rho_1}$
$\frac{(1-\rho)\bar{w}}{1-\rho_1-\rho_2}$
E[w1]
$\frac{(1-\rho)\bar{w}}{1-\rho_1}$
(1,1)
(0,1)
r_2
(1,0)
r_1
(0,0)

Figure 4.6 Mean Waiting Times Behavior Under Different Levels of Discrimination

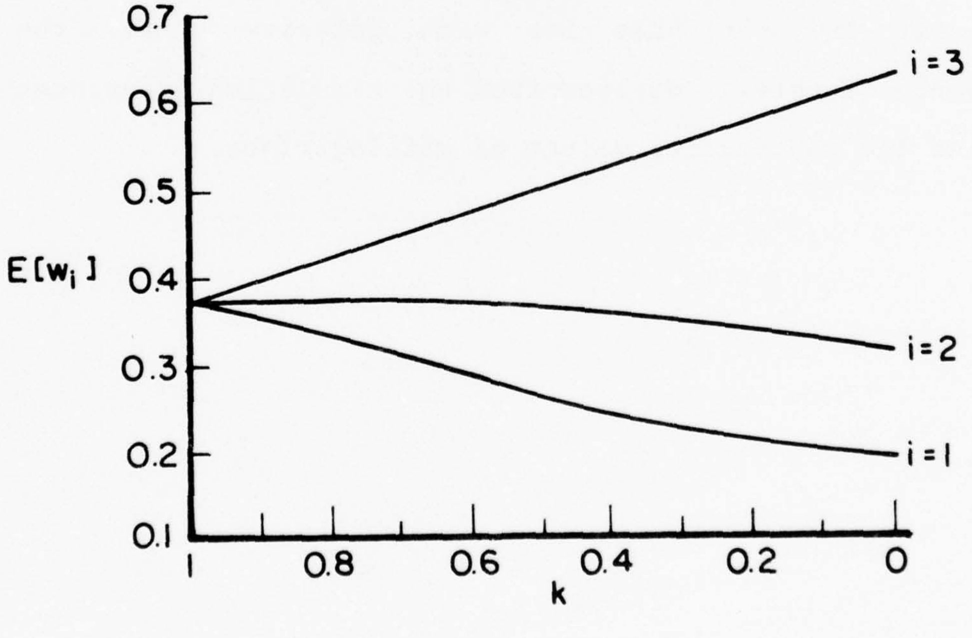


Figure 4.7 Mean Waiting Times Behavior Under Increasing Discriminations

It is interesting to note that in Figure 4.7, when k decreases below approximately 0.5, the difference between $E[W_3]$ and $E[W_2]$ increases faster than the difference between $E[W_2]$ and $E[W_1]$. This shows that when k is in this range, the discrimination between $E[W_3]$ and $E[W_2]$ caused by the decreasing of α_3 / α_2 is stronger than the discrimination between $E[W_2]$ and $E[W_1]$ caused by the decreasing of α_2 / α_1 by the same amount.

In Figure 4.8, the mean waiting times of the simulation experiments and the theoretical mean waiting times are compared. The solid lines represent theoretical waiting times for different priority classes, broken lines represent simulated waiting times, and dotted lines represent upper and lower 90% confidence limits of the simulated waiting times. It is seen that for each priority class, the confidence interval derived from the simulation experiment contains the theoretical values of waiting times.

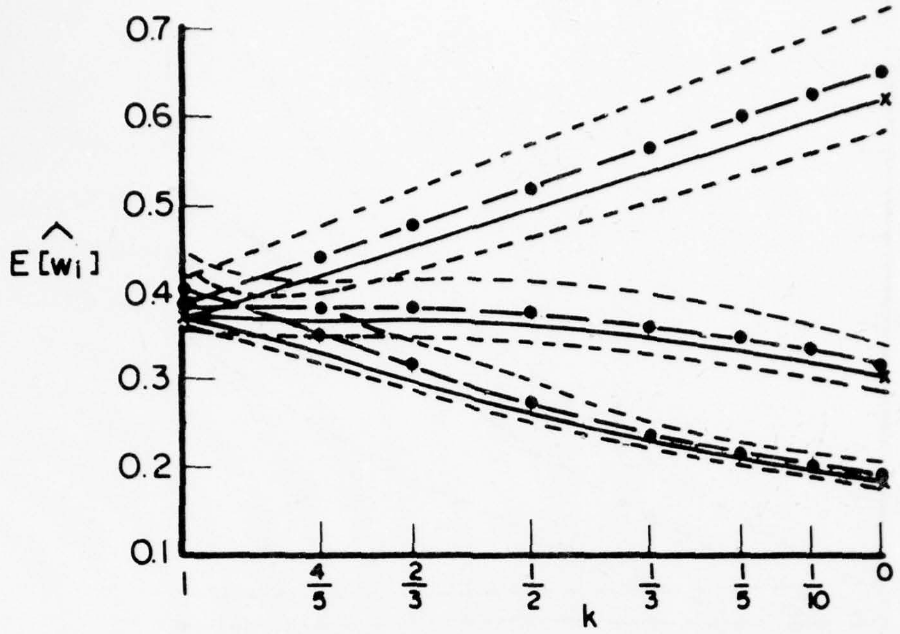


Figure 4.8 Point Estimates and Confidence Intervals of Mean Waiting Times

For the same simulation experiments, Figure 4.9 gives the variances of waiting times and their corresponding 90% confidence intervals. The marks "x" are the theoretical values of variances of the two limiting cases, FCFS and fixed priority scheduling. Note that each of the confidence intervals at the two limits also contain the theoretical values. These results show that as discriminations increase, the variance for each priority class changes strictly monotonically, and it either decreases or increases depending on whether the particular class is a high priority or a low priority class.

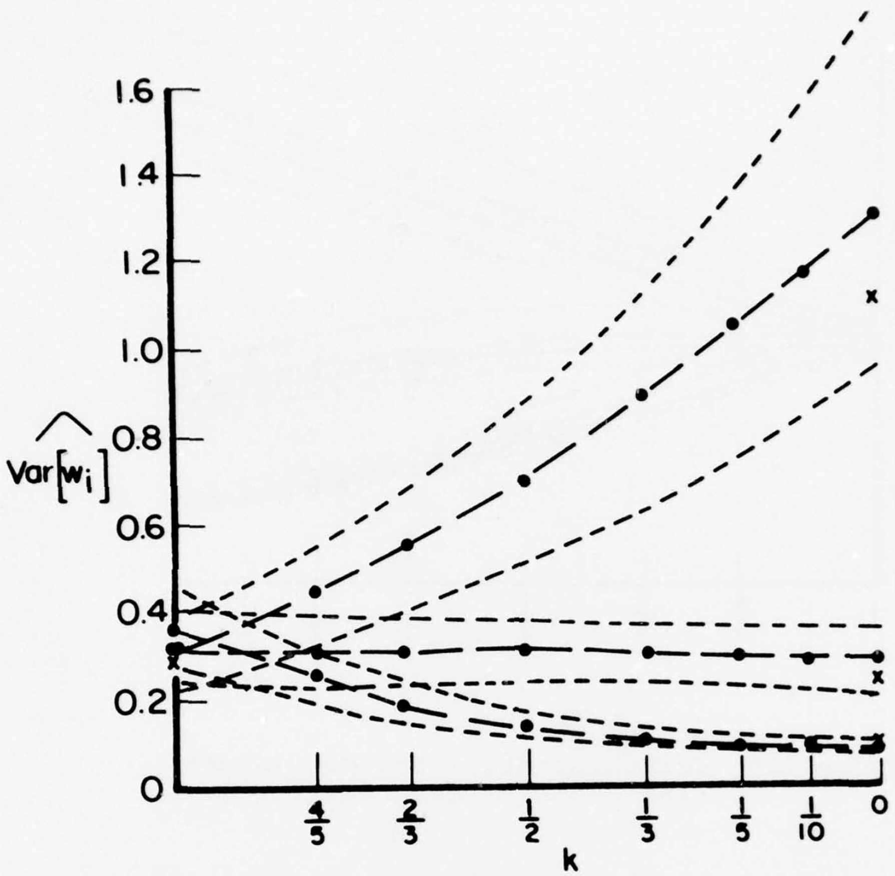


Figure 4.9 Variance Behavior of Waiting Times

Figure 4.10 gives the coefficient of variation (Cv) behavior for each class. We note that each curve of Cv[Wi] connecting $Cv[Wi]_{FCFS}$ and $Cv[Wi]_{FP}$ is approximately linear.

Experiments similar to the above were conducted for several different service time distributions. In all experiments, the variances of escalating priority scheduling were either strictly increasing or strictly decreasing between the variances of FCFS and fixed priority scheduling. Therefore, we conclude that we can determine the upper and lower bounds of the variance of any priority class in escalating priority scheduling from the variances of FCFS and fixed priority scheduling and obtain a "crude" estimate of its variance for a given parameter setting by using a coefficient of variation estimate from curves similar to those given in Figure 4.10.

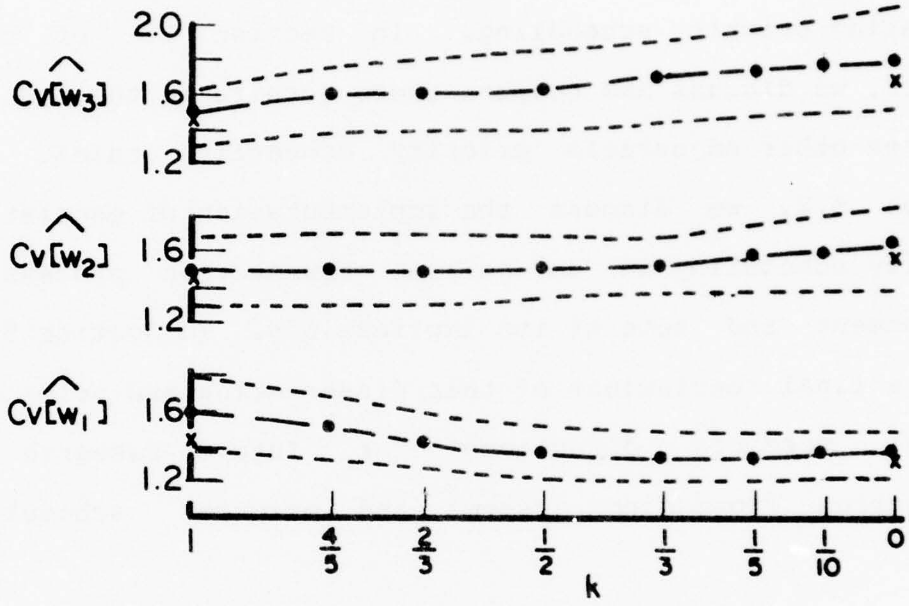


Figure 4.10 Coefficient of Variation Behavior of Waiting Times

Summarizing the results of our simulation study on the variances of waiting times in escalating priority scheduling, we find that as the discriminations increase, the variance of each class changes in the same direction (increases or decreases) as the mean waiting times, and has a value between the variances of FCFS and fixed priority scheduling.

CHAPTER 5

RELATED TOPICS AND CONCLUSIONS

5.0 Introduction

In the previous chapters, we have investigated in detail the behavior of M/G/l queueing systems under escalating priority scheduling. In Section 5.1 of this chapter, we discuss and compare these results to the results of three other adjustable priority scheduling rules. In Section 5.2, we discuss the implementation of escalating priority scheduling in an on-line transaction processing environment and some of its implications. In Section 5.3, we give final conclusions of this dissertation and point out several possible directions for future research on Transaction Processing Systems and priority scheduling rules.

5.1 Other Adjustable Priority Scheduling Rules

In this section, we will discuss and compare results of three other adjustable priority scheduling rules with escalating priority scheduling.

Specifically, we will discuss the adjustable priority scheduling rules with the following priority index functions:

(a) $q_i(t) = \beta_i(t-\tau_i)^r,$ $\qquad (\beta_i>0,\ r>0)$

(b) $q_i(t) = \gamma_i(t-\tau_i),$ $\qquad (\gamma_i<0)$

(c) $q_i(t) = \nu_i + (t-\tau_i)$.

Each of the above priority functions is the priority index assigned to a job from class i which arrived to the system at time τ_i. Whenever the service facility is available, the job with the highest instantaneous priority index $q_i(t)$ is taken into service. We will now discuss each of these scheduling rules in more detail.

(a) First, we consider the priority discipline in which a job's priority increases in proportion to some arbitrary power of its elapsed time waiting in the system. That is, at time t, the priority associated to a job in class i that arrived at time τ_i is assigned a value

$$q_i(t) = \beta_i\ (t - \tau_i)^r, \qquad (\beta_i > 0)$$

where r is a fixed positive constant and t ranges from τ_i until this job gets the service. The interaction of priority indices between jobs of different priority classes is illustrated in figure 5.1.

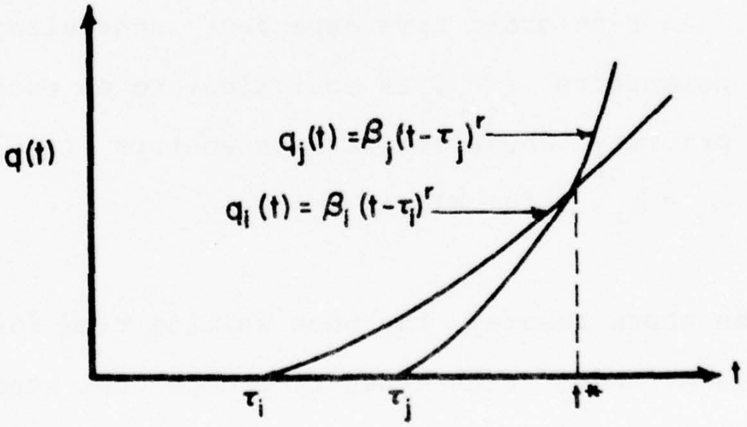


Figure 5.1 Interaction Between Two Jobs From Different Priority Classes

In this illustration, a job from class i arrives at time τ_i and gains priority proportional to the r-th power of the time it spent in the system. At time τ_j, another job from a higher priority class j enters the system, and gains its priority with a larger proportionality constant β_j. Thus, if the service facility becomes available between τ_i and t*, the job from class i will be taken into service. However, if the facility does not become free until t* or after, then the job from class j will be chosen for service.

This scheduling rule was first studied by Kleinrock and Finkelstein [KLEI67] and was called the r-th order time dependent scheduling. The following theorem, due to Kleinrock and Finkelstein, states that this scheduling rule is equivalent to escalating priority scheduling. Similar to his other results on escalating priority scheduling, this theorem is true for M/G/1 queueing systems.

THEOREM 5.1. An r-th order time dependent scheduling with parameters $\{\beta_i\}$ is equivalent to an escalating priority scheduling with parameters $\{\alpha_i\}$ when $\alpha_i = \beta_i^{\frac{1}{r}}$ for all i, $1 \le i \le n$.

From the above theorem, the mean waiting time for each priority class under r-th order time dependent scheduling can be obtained from (1.3.10) as:

$$E[W_i]^{(r)} = \frac{\overline{W} - \sum_{k=i+1}^{n} \left(1 - \left(\frac{\beta_i}{\beta_k}\right)^{\frac{1}{r}}\right) \rho_k E[W_k]}{1 - \sum_{k=1}^{i-1} \left(1 - \left(\frac{\beta_k}{\beta_i}\right)^{\frac{1}{r}}\right) \rho_k} .$$

The equivalence of the two scheduling shown in Theorem 5.1 reveals that they have the same behavior and the same feasible performance space of mean waiting times.

(b) Next, we discuss what we call "deescalating priority scheduling discipline" where each job's priority decreases linearly as it waits in the system. Specifically, we give to each priority class i a negative constant γ_i. At each instant t, the priority index $q_i(t)$ of a job from class i which arrived to the system at time τ_i is

$$q_i(t) = \gamma_i \ (t - \tau_i). \qquad (\gamma_i < 0)$$

Figure 5.2 illustrates the interaction between two jobs from different classes. In this example, at time τ_i, a job from class i arrives and loses priority at a rate β_i. At time τ_j, a different job from a lower priority class j arrives, and loses its priority at a faster rate γ_j. If the service facility becomes free between τ_j and t*, the job from class j will be served first. However, if the service facility does not become free until after t*, then the job from class i will be served prior to the job from class j.

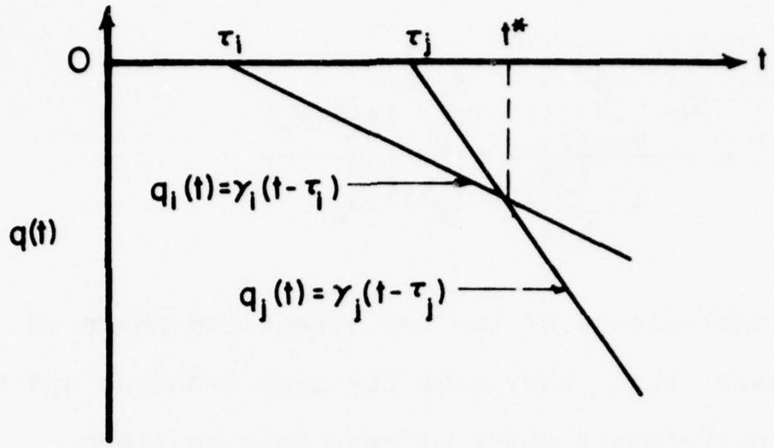


Figure 5.2 Interaction Between Priority Functions for Deescalating Priority Scheduling

The following theorem, due to Hsu [HSU70], describes the mean waiting times behavior of jobs from each priority class. This result was originally proved for M/M/1 queueing sytems. However, it holds true also for M/G/1 systems using the argument presented in Section 1.3.

THEOREM 5.2. In multiclass M/G/1 queueing systems under nonpreemptive deescalating priority scheduling, if $\rho < 1$ and we are given a set of negative parameters $\{\gamma_i\}$ such that $\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_n$, the mean waiting times for each priority class is:

$$E[W_i] = \frac{(1-\rho)\overline{W} + \sum_{k=1}^{i-1} (1 - \frac{\gamma_k}{\gamma_i})\rho_k E[W_k]}{(1-\rho) + \sum_{k=i+1}^{n} (1 - \frac{\gamma_i}{\gamma_k})\rho_k} .$$

The limits of this scheduling are given in the following theorem:

THEOREM 5.3. (a) When we set the γ_i's all equal, then deescalating priority scheduling becomes LCFS scheduling.

(b) In the limiting situation of deescalating priority scheduling when the parameters $\{\gamma_i\}$ are chosen such that $\gamma_i/\gamma_{i+1} \rightarrow 0$ $(1 \leq i \leq n-1)$, then we have a modified fixed priority scheduling rule such that within each priority class, a LCFS rule is followed.

For comparisons of waiting times, we give the following results due to Wishart [WISH60] and Durr [DURR69].

THEOREM 5.4. In multiclass M/G/1 queueing systems with $\rho < 1$, we have the following:

(a) Under LCFS scheduling, the mean and variance of waiting times are given by

$$E[W_{LCFS}] = \overline{W}$$

$$\mathrm{Var}[W_{LCFS}] = (1+\rho)\overline{W}^2/(1-\rho) + (\sum_{k=1}^{n} \lambda_k E[S_k^3])/3(1-\rho)^2 .$$

(b) Under modified fixed priority scheduling in which an LCFS rule is followed within each priority class, we have:

$$E[Wi] = (1-\rho)\overline{W} / (1-\sigma_{i-1})(1-\sigma_i) .$$

$$\mathrm{Var}[Wi] = \frac{\sum_{k=1}^{n} \lambda_k E[S_k^3]}{3(1-\sigma_{i-1})^2(1-\sigma_i)} + \frac{(1-\rho)\overline{W}(\sum_{k=1}^{i} \lambda_k E[S_k^2])}{(1-\sigma_{i-1})(1-\sigma_i)^3}$$

$$+ \frac{(1-\rho)\overline{W}\{\sum_{k=1}^{i-1} \lambda_k E[S_k^2] - (1-\rho)\overline{W}\}}{(1-\sigma_{i-1})^2(1-\sigma_i)^2} .$$

Comparing the above formula with those in Theorems 1.2 and 1.3, we see that mean waiting times under either FCFS or LCFS scheduling are the same. Similarly, mean waiting time for each priority class remains the same no matter whether an FCFS or an LCFS scheduling is followed within the same priority class. However, variances under LCFS scheduling are larger than that under FCFS scheduling, and in fixed priority scheduling, variance of waiting times for each class is larger if LCFS rule is used instead of the FCFS rule within the same priority class.

It is most likely that an algorithm can be found for determining the values of $\{\gamma_i\}$ for deescalating priority scheduling using a method similar to that given in Theorem 2.4 for escalating priority scheduling. We note that the feasible performance space of mean waiting times for these two scheduling rules are identical. However, for the same performance of mean waiting times, deescalating priority scheduling will have larger variances of waiting times.

(c) We now discuss the scheduling rule for the following priority function:

$$q_i(t) = \nu_i + (t - \tau_i).$$

In this scheduling rule, higher priorities are given to the classes with larger values of the ν_i's, because for two jobs arriving to the system at the same time, the one from the class with larger value of ν_i will have a larger priority index. Also, note that in this rule, the parameters $\{ \nu_i \}$ are not required to be positive.

Now, if the quantity $(\tau_i - \nu_i)$ is the due date of a job that arrives to the system at time τ_i , then the selection of tne job with the largest $q_i(t) = \nu_i + (t - \tau_i) = t - (\tau_i - \nu_i)$ at any instant of time t is equivalent to selecting the job with smallest values of $\tau_i - \nu_i$, i.e., the job with the smallest due date. Thus, this scheduling is called the earliest due date (EDD) scheduling when the ν_i's are chosen in this way and we will restrict ourselve to this interpretation in the remaining discussion of this priority function.

Figure 5.3 illustrates the priorities of two jobs from different priority classes. It is noted that no interaction of priorities exists between the two jobs under this scheduling rule.

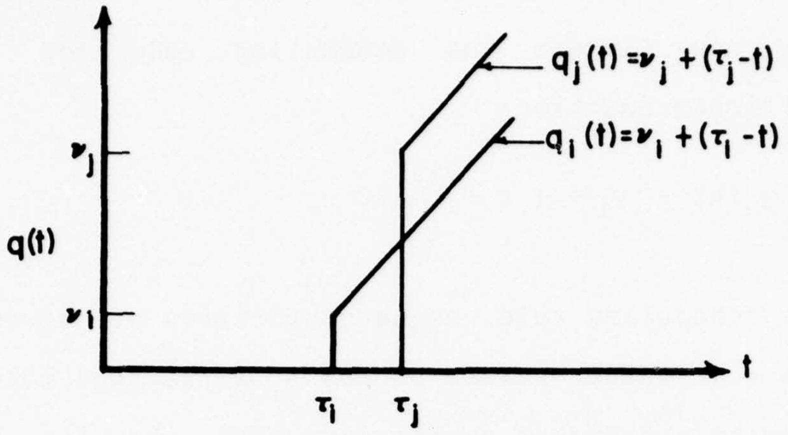


Figure 5.3 Priority Indices of Two Jobs In Earliest Due Date Scheduling

In the literature discussing this scheduling rule, instead of defining the priority function using the parameters $\{ \nu_i \}$, a set of constants $\{ u_i \}$ called urgency numbers are used. In this scheme of defining priorities, when a job from class i arrives at t_i, it is assigned a priority number t_i+u_i. When the server becomes available, it chooses the job having the minimum value of t_i+u_i. It can be readily seen that the two schemes of selection are equivalent, when $u_i = -\nu_i$ for each i.

The earliest due date scheduling discipline was first studied by Jackson [JACK60], who made the following conjecture: In multiclass M/G/1 queueing systems under nonpreemptive EDD scheduling, we have

$$\lim_{\rho \to 1} (\nu_i + E[W_i])/(\nu_j + E[W_j]) = 1. \qquad (1 \le i \le n, 1 \le j \le n)$$

The above conjecture, known as the Jackson's Conjecture, was proved by Goldberg [GOLD76].

It is interesting to note the similarity of this result with Theorem 4.3: In escalating priority scheduling, when the priority of a job from class i is $\alpha_i W_i$, we have the following:

$$\lim_{\rho \to 1} (\alpha_i E[W_i])/(\alpha_j E[W_j]) = 1. \qquad (1 \leq i \leq n,\ 1 \leq j \leq n)$$

Thus, Theorem 4.3 can be thought of as the multiplicative version of Jackson's Conjecture for escalating priority scheduling systems.

Next, we recall from Theorem 4.2 that under escalating priority scheduling, when $\rho < 1$, we have:

$$(4.1.3) \qquad \alpha_1 E[W_1] \geq \alpha_2 E[W_2] \geq \ldots \geq \alpha_n E[W_n].$$

Similar to this result, Goldberg proved the following: In nonpreemptive EDD scheduling, for $\rho < 1$,

$$\nu_1 + E[W_1] \geq \nu_2 + E[W_2] \geq \ldots \geq \nu_n + E[W_n].$$

Goldberg also proved that the limits of EDD scheduling are FCFS and fixed priority scheduling, and derived the following formula for mean waiting times of each priority class:

$$E[Wi] = \overline{W} + \sum_{k=1}^{i-1} \rho_k \int_0^{\nu_k-\nu_i} Pr\{Wi>t\}dt - \sum_{k=i+1}^{n} \rho_k \int_0^{\nu_i-\nu_k} Pr\{W >t\}dt.$$

Unfortunately, the above formula can not be directly solved to obtain $\{ \nu_i \}$ given a set of specified and feasible mean waiting times.

We conclude from the above discussions that adjustable priority scheduling rules with one set of adjustable parameters can provide several different types of waiting times behaviors. It may be possible to obtain the same mean waiting times for the different priority classes using different adjustable scheduling rules, but the variances of the waiting times may be different. To obtain control of behaviors of both the mean and variance of waiting times, scheduling rules with more than one set of parameters may be needed. Thus, for instance, if we use a scheduling rule with priority function defined as

$$q_i(t) = \nu_i + \alpha_i (t-\tau_i),$$

then the behavior of waiting times may be able to be controlled to a greater extent.

5.2 Escalating Priority Scheduling for TP Systems

In Chapter 1 of this dissertation, we pointed out that transaction processing systems commonly use fixed priority scheduling with the result that transactions from lower priority classes frequently have excessively long response times while transactions from higher priority classes may have response times that are faster than necessary. This problem motivated us to study escalating priority scheduling to determine if it would be appropriate for use in TPS. In this section, we will discuss the use of escalating priority scheduling in the class of TPS that can be modeled as multiclass M/G/1 queueing models.

The use of escalating priority scheduling in TPS will allow the designer of TPS to select a set of values for the control parameters $\{ \alpha_i \}$ to provide any desired discrimination between the transaction classes from FCFS to fixed priority scheduling. This capability should help eliminate undesired long response times for transactions from low priority classes when transactions from high priority classes are receiving faster response times than necessary. The algorithm presented in Chapter 2 allows the designer to obtain a set of $\{ \alpha_i \}$ for any set of feasible expected response times for the various transaction classes.

If a designer of TPS desires to select the set of control parameters $\{ \alpha_i \}$ of escalating priority scheduling to optimize the cost of a TPS using some function of the expected waiting times, a set of feasible expected waiting times can be obtained by using one of the optimizing algorithm presented in Chapter 3. This set of waiting times then can be used in the algorithm in Chapter 2 to obtain the required values of $\{ \alpha_i \}$. This procedure requires the designer to choose an appropriate cost function and to obtain the appropriate unit cost estimates. One should note that this optimization procedure does not take into account other aspects of response times that may be important such as the variances of response times or the system response under saturated congestion.

When the transaction workload becomes very heavy, one possible disadvantage of using escalating priority scheduling in TPS is that the response times of all classes of transactions increase together. This is quite different from what occurs when using fixed priority scheduling where only the low priority classes response times increase under a heavy load of transactions. This can be avoided if the designer of the TPS uses MEFP scheduling. However, one significant advantage of using escalating priority scheduling is that for small changes in transaction traffic, the relative response timesfor different transaction classes stay approximately the same.

The variances of response times are important in many TPS because they can affect the productivity of the human users. Our results of escalating priority scheduling show that the variances of response times for lower priority classes are much smaller than in fixed priority scheduling. However, the variances of the higher priority classes will increase.

We have now stated that escalating priority scheduling is applicable and that there are many advantages for using it in TPS instead of fixed priority scheduling. However, any scheduling rule used in TPS must require only a small amount of computation time for each arriving transaction and for determining the next transaction for service. This is because the transaction traffic is large and the processing time required for each transaction is usually very small. Therefore, the overhead for handling each transaction must be small.

We now discuss the implementation of the escalating priority scheduling rule. First, we note that under this scheduling rule, jobs from the same priority class will be selected strictly according to their arrival times. This is because jobs from the same class have the same priority increasing rate. Thus, the job that has been waiting in the system for the longest period will have the highest priority. Therefore, if we maintain a separate queue for each priority class, and when a job arrives, we file it into the end of the queue corresponding to its priority class, then only the first job of each queue can qualify as the candidate for selection. This reduces the number of possible comparisons in selecting the next job for service to n, the number of different priority queues.

The above observation also has the following implication: Even though the priority indices of the waiting jobs increase dynamically, there is no need to update the priority indices of all jobs every time the server becomes available. We only need to register the arrival time for each job when it enters to the queue. When the server is ready for next service, we then take measures of the elapsed time that each prospective candidate (the first job in each queue) has waited in the queue, and multiply them respectively by their corresponding priority increasing rates to calculate their priority indices. The job with highest priority index is then removed from the queue and is put into service.

This shows that escalating priority scheduling can be efficiently implemented in transaction processing systems with very little time required for overhead information processing.

5.3 Conclusions and Future Research

The purpose of this dissertation was to investigate escalating priority scheduling and determine if it is appropriate for use in TPS. The major result of our investigation of escalating priority scheduling was the development of a two-stage algorithm for determining the values of the control parameters to optimize various cost functions. This two-stage algorithm allows a designer of

TPS to obtain a set of control parameters for escalating priority scheduling in TPS to optimize certain types of cost functions that should provide better response times to users of TPS than currently being provided by fixed priority scheduling. Thus this scheduling is applicable to TPS and should provide better utilization of the system resources.

This dynamic priority scheduling rule was first proposed by Kleinrock in 1964. His major contribution, in addition to proposing the rule, was in developing an algorithm to obtain the values of mean waiting times given the control parameters $\{\alpha_i\}$. The major new results for this scheduling rule presented in this dissertation are: (1) the derivation of the feasible performance space for mean waiting times, (2) the development of an algorithm to obtain a set of $\{\alpha_i\}$ given a set of feasible mean waiting times {E[Wi]}, (3) optimizing algorithms to obtain the {E[Wi]} to be used in obtaining the $\{\alpha_i\}$ for minimizing various cost functions, and (4) proving a multiplicative version of Jackson's conjecture in saturated congestion.

Other results presented for the escalating priority scheduling rule are that the variance of waiting times for each class either strictly increases or strictly decreases between the variances of FCFS and fixed priority scheduling for various parametric settings; at saturated congestion the mean response times of all classes are extremely large; and the relative response times for each class have little change for changes in the arrival rates.

There are a number of interesting problems that need to be studied for further understanding of scheduling transaction processing systems. The question that arises first is: What is the behavior of TPS using escalating priority scheduling when there is only a small number of users instead of a large number? This requires investigation of the multiclass M/G/1 queueing model with a finite arrival population under escalating priority scheduling. This may be difficult because the arrival rate changes as the number of transactions that are waiting for or being processed increases or decreases.

The second question that arises is: What happens in escalating priority scheduling if the arrivals are not Poisson? This could occur if the number of transaction arrivals for a given class was not large.

A third area for research is to investigate and compare other adjustable scheduling rules. Of particular interest is: What are the relationships between the mean and variance of different adjustable scheduling rules? Can the same mean response times be achieved for different scheduling rules but with different variances?

Another large area for research is when multiprogramming is allowed in TPS. In this case, queueing network models must be used. Here, the various arrival processes, service time distributions and different scheduling rules with optimization need to be researched.

Finally, it would be valuable to collect some "live data" on various systems for analysis and to perform various types of experiments using different scheduling rules. Some measurements of interest are the arrival processes, the service time requests, waiting time distributions, and mean response times. This would increase the understanding of the behavior of transaction processing systems.

BIBLIOGRAPHY

[ACZE60] Aczel, M. A., "The Effect of Introducing Priorities," Operations Research, 8, 730-733 (1960)

[AGRA75] Agrawala, A. K. and J. M. Mohr, "On Superposition of Stochastic Point Processes and Their Application to Computer System Modeling," Technical Report, University of Maryland, 1975.

[ANDE74] Anderson, H. A. Jr. and R. G. Sargent, "Investigation into Scheduling for an Interactive Computing System," I.B.M. Journel of Research and Development, 18, 125-137 (1974)

[BAKE74] Baker, K. R., Introduction to Sequencing and Scheduling, John Wiley, (New York) 1974.

[BALA70] Balachandran, K. R., "Parametric Priority Rules: An Approach to Optimization in Priority Queues," Operations Research, 18, 526-540 (1970)

[BOOT72] Booth, G., "Transaction Processing Systems," Data Management, 14-16 (1972)

[BROS63] Brosh, I. and P. Noar, "On Optimal Disciplines in Priority Queueing," Bull. Inst. Statist. Math., 40, 593-607 (1963)

[CARB68] Carbonell, J., J. Elkind and R. Nickerson "On the Psychological Importance of Time in a Time-Sharing System", Human Factors, 10, 135-142 (1968)

[COBH54] Cobham, A., "Priority Assignment in Waiting Line Problems," Operations Research, 2, 70-76 (1954)

[CONW67] Conway, R. W., W. L. Maxwell, and L. W. Miller, Theory of Scheduling, Addison-Wesley (Reading, Mass.), 1967.

[CRAB73] Crabill, T. B., D. Gross, and M. J. Magazine, "A Classified Bibliography of Research on Optimal Design and Control of Queues," Operations Research, 25, 219-232 (1977)

[DAVE74] Davenport, R. A., "Design of Transaction-Oriented Systems Employing a Transaction Monitor," Proceedings ACM Annual Conference, 222-230 (1974)

[DURR69] Durr, L., "A Single-Server Priority Queueing System with General Holding Times, Poisson Input and Reverse-Order of Arrival Queueing Discipline" Operations Research, 17, 351-358 (1969)

[EADE77] Eade, D. J., P. Homan, and J. H. Jones, "CICS/VS and Its Role in Systems Network Architecture," IBM System Journal, 16, 258-286 (1977)

[FIFE65] Fife, D. W., "Scheduling With Random Arrivals and Linear Loss Functions," Management Science, 11, No. 3, 429-437 (1965)

[GERA76] Gerami, C. R., T. R. Shields and R. J. Weiland, "Transaction Queueing and Cylinder Logic Access in the Time, Inc. Magazine/ Book/ Record System," AFIPS Conference Proceedings, 45, 883-888 (1976)

[GERK74] Gerke, S. P., "Performance Projection and Evaluation for a Transaction-Oriented System," Symposium on the Simulation of Computer System II, 77-83 (1974)

[GOLD77] Goldberg, H., "Analysis of the Earliest Due Date Scheduling Rule in Queueing Systems," Mathematics of Operations Research, 2, 145-154 (1977)

[HAJI71] Haji, R., and G. F. Newell, "Optimal Strategies for Priority Queues with Nonlinear Costs of Delay," SIAM Journal of Applied Mathematics, 20, 224-240 (1971)

[HIRC75] Hirchfeld, L. J., "Design Methodology for Transaction Processing Systems," Ph. D. Dissertation, University of Pennsylvania, 1975.

[HOAR72] Hoare, C. A. R., "A General Conservation Law for Queueing Disciplines," Information Processing Letters, 2, 82-85 (1972)

[HOLT71] Holtzman, J. M., "Bounds for a Dynamic Priority Queue," Operations Research, 19, 461-468 (1971)

[HONE73] Honeywell Information Systems, Series 600/6000 GCOS Transaction Processing System User's Guide, Order No. DA82, 1973.

[HOOK71] Hooke, J. A., and N. U. Prabhu, "Priority Queues in Heavy Traffic," Opsearch, 8, 1-9 (1971).

[HSU70] Hsu, J., "A Continuation of Delay-Dependent Queue Disciplines," Operations Research, 18, 733-738 (1970)

[IBM73] International Business Machines, *Customer Information Control System (CICS) General Information Manual*, GH20-1028-4, 1973.

[JACK60] Jackson, J. R., "Some Problems in Queueing with Dynamic Priorities," *Naval Research Logistics Quarterly*, *7*, 235-249 (1960)

[JACK61] Jackson, J, R., "Queues with Dynamic Priority Discipline," *Management Science*, *8*, No. 1, 18-34 (1961)

[JACK62] Jackson, J. R., "Waiting-Time Distributions for Queues with Dynamic Priorities," *Naval Research Logistics Quarterly*, *9*, 31-36 (1962)

[JAIS68] Jaiswal, N. K., *Priority Queues*, Academic Press (New York), 1968.

[JESS76] Jessen, T. D., "Transaction Oriented Minicomputer Allows Flexible Design of the Controlled Materials Information System," *NTIS Report*, UCRL-77948, 1976.

[KEND51] Kendall, D. G., "Some Problems in the Theory of Queues," *J. Roy. Statist. Soc. Ser. B*, *13*, 151-185 (1951)

[KING62] Kingman, J. F. C., "The Effect of the Queue Discipline on Waiting Time Variance," *Proceedings of the Cambridge Philosophical Society*, *58*, 163-164 (1962)

[KLEI64] Kleinrock, L., "A Delay Dependent Queue Discipline," *Naval Research Logistics Quarterly*, *11*, 329-341 (1964)

[KLEI65] Kleinrock, L., "A Conservation Law for a Wide Class of Queueing Disciplines," *Naval Research Logistics Quarterly*, 12, 181-192 (1965)

[KLEI66] Kleinrock, L., "Queueing with Strict and Lag Priority Mixtures," *Proceedings of the 4th International Conference on Operational Research*, Boston, Mass. K-I-46 to K-I-67 (1966)

[KLEI67] Kleinrock, L., and R. P. Finkelstein, "Time Dependent Priority Queues," *Operations Research*, *15*, 104-116 (1967)

[KLEI75] Kleinrock, L., *Queueing Systems, Vol. 1: Theory*, Wiley Interscience, (New York), 1975.

[KLEI76] Kleinrock, L., Queueing Systems, Vol. 2: Computer Applications, Wiley Interscience, (New York), 1976.

[KNIG72] Knight, J. R., "A Case Study: Airlines Reservation Systems," Proceedings of the IEEE, 11, 1423-1431 (1972)

[KUHN51] Kuhn, H. W. and A. W. Tucker, "Nonlinear Programming," Proceedings Second Berkeley Symposium on Mathematical Statistics and Probability, University of Berkeley Press, 1951.

[LEFK74] Lefkovitz, D., Data Management for On - Line Systems, Hayden Book Company, Inc., 1974.

[LEFK75] Lefkovitz, D., and L. J. Hirchfeld, "Transaction Processing via Processor Network," unpublished paper, Department of Computer Science, University of Pennsylvania, 1975.

[LEWI76] Lewis, P. A. W., and G. S. Shedler, "Statistical Analysis of Non-Stationary Series of Events in a Data Base System," IBM Journal of Research and Development, 20, 465-482 (1976)

[LIPS64] Lipschutz, S., Theory and Problems of Set Theory and Related Topics, Schaum Publishing Co., (New York), 1964.

[LITT61] Little, J. D., "A Proof for the Queueing Formula $L = \lambda W$," Operations Research, 9, 383-387 (1961)

[MARC74] Marchand, M. G., "Priority Pricing," Management Science, 20, 1131-1140 (1974)

[MATH77] Matheny C. S. and D. C. Peak, Teleprocessing Monitors: A Technical Analysis, published by Q.E.D. Information Sciences, Inc., (Wellesley, Mass.) 1977.

[MCGE77] McGee, W. C., "The Information Management System IMS / VS Part V: Transaction Processing Facilities," IBM System Journal, 16, 148-168 (1977)

[MCKE73] McKee, D. J., "How Transaction Cost Declines as Data Networks Get Larger," Data Communications Systems, Sept. 1975.

[MILL68] Miller, R. B., "Response Time in Man-Computer Conversational Transactions," Proceedings Fall Joint Computer Conference, 267-277 (1968)

[OSBO75] Osborn, R. A., W. P. Bain and P. J. Perring, "SPG - A Programming System for Commercial Transactions Processing," The Computer Journal, 290-297 (1975)

[PERR61] Perry, M. N., and W. R. Plugge, "American Airlines SABRE Electronic Reservation System," AFIPS Conference Proceedings, Western Joint Computer Conference, 19, 593-601 (1961)

[PANW77] Panwalker, S. S., and W. Iskander, "A Survey of Scheduling Rules," Operations Research, 25, 45-61 (1977)

[RADC75] Rome Air Development Center, "TPAP Operating System Executive," Technical Report, RADC-TR-75-212, 1975.

[ROSB76] Rosberg, Z., and I. Adiri, "Multilevel Queues with External Priorities," Journal of the Association for Computing Machinery, 23, 680-690 (1976)

[RUSC78] Ruschitzka, M., "Performance Evaluation of Nonpreemptive Response-Ratio Schedulers," AFIPS Conference--Proceedings, 47, 473-481 (1978)

[SARG76] Sargent, R., "Statistical Analysis of Simulation Output Data," Proceedings of the ACM Symposium on the Simulation of Computer Systems IV, 39-50 (1976)

[SARZ77] Sarzotti, A., "Transactional Terminal System on Micro-Processor: A Method for Identifying and Modelling Overall Performance," Proceedings of the 1977 Sigmetrics / CMG VIII Conference on Computer Performance, Edited by R. Morrison, Association of Computiong Machinery, New York 1977.

[SCHE76] Schember, K., "Optimal Design of Files for Transaction-Oriented Data Base Systems," Ph. D. Dissertation, Texas A & M University, 1976.

[SCHW77] Schwandt, J., "An Approach to Use Evaluation Nets for the Performance Evaluation of Transaction-Oriented Business Computer Systems," Computer Performance, Edited by K. M. Chandy and M. Reiser, North-Holland (New York) 1977.

[SCHR68] Schrage, L., "A Proof of the Optimality of the Shortest Remaining Processing Time Discipline," Operations Research, 16, 687-690 (1968)

[SCHR70] Schrage, L., "An Alternative Proof of a Conservation Law for the Queue G/G/1," Operations Research, 18, 185-187 (1970)

[SCHR74] Schrage, L., "Optimal Scheduling Disciplines for a Single Machine Under Various Degree of Information," Working Paper, Graduate School of Business, University of Chicago, 1974.

[SEVC74] Sevcik, K., "Scheduling for Minimum Total Cost Using Service Time Distribution," Journal of the Association for Computing Machinery, 21, 66-75 (1974)

[SIWI77] Siwiec, J. E., "A High Performance DB/DC System," IBM System Journal, 16, 169-195 (1977)

[STER75] Stern, H. C., "Cost / Benefit Analyusis of Transaction Processing Systems," ACM Computer Science Conference, Washington, D. C., 1975.

[TAHA76] Taha, H., OPerations Research: An Introduction, Macmillan Publishing Co., (New York) 1976.

[TAKA64] Takacs, L., "Priority Queues," Operations Research, 12, 63-74 (1964)

[TAMB68] Tambouratzis, D. G., "On the Property of the Variance of the Waiting Time of a Queue," Journal of Applied Probability, 5, 702-703 (1968)

[THEI75] Theirauf, R. J., Systems Analysis and Design of Real-Time Management Information System, Prentice Hall, 1975.

[WISH60] Wishart, D. M. G., "Queueing Systems in Which the Discipline Is Last-Come-First-Served," Operations Research, 8, 591-599 (1960)

[WOLF70] Wolf, R. W., "Work Conserving Priorities," Journal of Applied Probability 7, 327-337 (1970)

[WOOD78] Wood, D. K., and R. G. Sargent, "An Introduction to Transaction Processing Systems," Large Scale Information Systems, Vol. 1, Technical Report RADC-TR-78-43, Rome Air Development Center, 1978. (A054 942)

APPENDIX A

A SIMULATION MODEL FOR ESCALATING PRIORITY SCHEDULING

In order to study the variance behavior of waiting times in M/G/1 queueing systems under escalating priority scheduling, a simulation model using SIMSCRIPT II.5 was built. This model is a 3-class priority queueing model with each of the three priority classes having the same common characteristics: $\lambda_i = 0.5$, $\mu_i = 2.5$ and $\rho_i = 0.2$.

Four sets of experiments were conducted, each using a different service time distribution. The same service time distribution was used for each of the three classes of transactions in each of the sets of experiments to hopefully have the results reflect only the variance behavior. The service time distributions used were the following:

Distributions	Coefficient of Variations
Constant	0
Erlang-4	0.5
Exponential	1.0
Hyperexponential	2.0

The values of the control parameters were determined using $(\alpha_1, \alpha_2, \alpha_3) = (a, ak, ak^2)$ with k decreasing from one towards zero and $a > 0$. This resulted in the relative discriminations between classes 1 and 2 and classes 2 and 3 being the same for each setting of k because $\alpha_1/\alpha_2 = \alpha_3/\alpha_2 = 1/k$. The specific set of values of k used for each set of experiments were k=1, 4/5, 2/3, 1/2, 1/3, 1/5, 1/10, and 1×10^{-10}. The setting k equals one corresponds to first-come-first-serve scheduling and the setting k equals 1×10^{-10} approximates fixed priority scheduling.

The regenerative method of analysis was used to collect and analyze the simulation data [SARG76]. The regenerative point selected was the beginning of a busy cycle. The number of cycles used were 2,500 for the set of experiments having constant service time distributions and 3,000 for the other service time distributions. Data were collected on the first and second moments of the waiting times.

For each service time distribution, using the sample mean of waiting times, we constructed a 90% confidence interval of mean waiting times. In all cases, the theoretical values were contained in the confidence intervals. This validated the model. The data for the set of experiments using Erlang-4 service time distributions are shown in Figure 4.8.

Using the sample second moments of waiting times and theoretical values of mean waiting times, we calculated the point estimates and confidence intervals for the variances of waiting times. For the cases when k=1 and $k=1 \times 10^{-10}$, the theoretical values of variances were obtained from (4.3.1) and (4.3.2), respectively. Again the 90% confidence intervals contain the theoretical values. This provided additional evidence that the model is valid. The data for the set of experiments using Erlang-4 service time distributions are given in Figure 4.9.

The point estimates and confidence intervals of the coefficient of variance ($Cv[W_i]$) were calculated for all experiments using the point estimates of the means and variances of the waiting times. To calculate the confidence intervals of $Cv[W_i]$, we used the following argument, assuming independence, from probability theory:

Given $Pr\{a \leq x \leq b\} = P\%$ and $Pr\{c \leq y \leq d\} = Q\%$,

then $Pr\{a/d \leq x/y \leq b/c\} < PQ\%$.

Using this argument and the estimates of the first and second moments, 81% confidence intervals were constructed for all of our experiments. Figure 4.10 contains the point and intervals estimates for the set of experiments using the Erlang-4 service time distributions.

The results of the four set of experiments were very similar except that the variances of the waiting times increased as the coefficient of variations of service times increased from zero to two as was expected. The means and variances of waiting times for each class of transactions strictly increased or decreased as k decreased from one to zero, depending on whether it was a high or low priority class. Furthermore, the coefficients of variations were similar for each set of experiments in that the coefficient of variations remained nearly constant for each priority class as k decreased from one to zero.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL
1776-1976

79